Y4.B22/1:99-65

2

THE FEDERAL RESERVE BANK OF NEW YORK DISCOUNT WINDOW ADVANCE OF $22.6 BILLION EXTENDED TO THE BANK OF NEW YORK

COMPLETED
625

Mf 157

HEARING

BEFORE THE

SUBCOMMITTEE ON DOMESTIC MONETARY POLICY

OF THE

COMMITTEE ON
BANKING, FINANCE AND URBAN AFFAIRS
HOUSE OF REPRESENTATIVES

NINETY-NINTH CONGRESS

FIRST SESSION

DECEMBER 12, 1985

Serial No. 99-65

Printed for the use of the
Committee on Banking, Finance and Urban Affairs

ORIGINAL

THE FEDERAL RESERVE BANK OF NEW YORK DISCOUNT WINDOW ADVANCE OF $22.6 BILLION EXTENDED TO THE BANK OF NEW YORK

HEARING

BEFORE THE

SUBCOMMITTEE ON DOMESTIC MONETARY POLICY

OF THE

COMMITTEE ON
BANKING, FINANCE AND URBAN AFFAIRS
HOUSE OF REPRESENTATIVES

NINETY-NINTH CONGRESS

FIRST SESSION

DECEMBER 12, 1985

Serial No. 99-65

Printed for the use of the
Committee on Banking, Finance and Urban Affairs

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON : 1986

For sale by the Superintendent of Documents, Congressional Sales Office
U.S. Government Printing Office, Washington, DC 20402

57-008 O

HOUSE COMMITTEE ON BANKING, FINANCE AND URBAN AFFAIRS

FERNAND J. ST GERMAIN, Rhode Island, *Chairman*

HENRY B. GONZALEZ, Texas
FRANK ANNUNZIO, Illinois
PARREN J. MITCHELL, Maryland
WALTER E. FAUNTROY, District of Columbia
STEPHEN L. NEAL, North Carolina
CARROLL HUBBARD, Jr., Kentucky
JOHN J. LaFALCE, New York
STAN LUNDINE, New York
MARY ROSE OAKAR, Ohio
BRUCE F. VENTO, Minnesota
DOUG BARNARD, Jr., Georgia
ROBERT GARCIA, New York
CHARLES E. SCHUMER, New York
BARNEY FRANK, Massachusetts
BUDDY ROEMER, Louisiana
RICHARD H. LEHMAN, California
BRUCE A. MORRISON, Connecticut
JIM COOPER, Tennessee
MARCY KAPTUR, Ohio
BEN ERDREICH, Alabama
SANDER M. LEVIN, Michigan
THOMAS R. CARPER, Delaware
ESTEBAN E. TORRES, California
GERALD D. KLECZKA, Wisconsin
BILL NELSON, Florida
PAUL E. KANJORSKI, Pennsylvania
BART GORDON, Tennessee
THOMAS J. MANTON, New York
JAIME B. FUSTER, Puerto Rico

CHALMERS P. WYLIE, Ohio
STEWART B. McKINNEY, Connecticut
JIM LEACH, Iowa
NORMAN D. SHUMWAY, California
STAN PARRIS, Virginia
BILL McCOLLUM, Florida
GEORGE C. WORTLEY, New York
MARGE ROUKEMA, New Jersey
DOUG BEREUTER, Nebraska
DAVID DREIER, California
JOHN HILER, Indiana
THOMAS J. RIDGE, Pennsylvania
STEVE BARTLETT, Texas
TOBY ROTH, Wisconsin
ROD CHANDLER, Washington
AL McCANDLESS, California
JOHN E. GROTBERG, Illinois
JIM KOLBE, Arizona
J. ALEX McMILLAN, North Carolina

SUBCOMMITTEE ON DOMESTIC MONETARY POLICY

WALTER E. FAUNTROY, District of Columbia, *Chairman*

STEPHEN L. NEAL, North Carolina
DOUG BARNARD, Jr., Georgia
CARROLL HUBBARD, Jr., Kentucky
BUDDY ROEMER, Louisiana
JIM COOPER, Tennessee
THOMAS R. CARPER, Delaware
BEN ERDREICH, Alabama

BILL McCOLLUM, Florida
JOHN HILER, Indiana
JIM LEACH, Iowa
STEVE BARTLETT, Texas
THOMAS J. RIDGE, Pennsylvania
ROD CHANDLER, Washington

HOWARD LEE, *Staff Director*

CONTENTS

Page
Notice of hearing and witness list 4

WITNESSES

Bacot, J. Carter, chairman and chief executive officer, The Bank of New York; accompanied by Deno Papageorge, senior executive vice president, chief financial officer; and Sheldon Harrison, senior executive vice president, chief operations officer 84
  Prepared statement 89
Corrigan, E. Gerald, president, Federal Reserve Bank of New York 23
  Prepared statement 31
Volcker, Hon. Paul A., Chairman, Board of Governors of the Federal Reserve System 5
  Prepared statement 9

MATERIAL SUBMITTED FOR INCLUSION IN THE RECORD

Bacot, J. Carter:
  Annual Report of The Bank of New York for the year 1984 135
  "Consolidated Reports of Condition and Income for A Bank With Domestic and Foreign Offices—FFIEC 031", document 109
  Excerpt from Moody's Bank and Finance Manual 134
Corrigan E. Gerald, response to questions of Chairman Fauntroy 60, 63

APPENDIX

"A Computer Snafu Snarls the Handling of Treasury Issues," article from the Wall Street Journal, dated November 25, 1985 563
"A Hard Day's Night for the Bank of New York," article from the American Banker, dated November 27, 1985 566
Bacot, Carter J., letter to Hon. Walter E. Fauntroy dated January 22, 1986, pertaining to a detailed review of its Government Securities Clearance Operation 207
"Bank of New York Computer Failure Disrupts U.S. Securities Market," article from the American Banker, dated November 25, 1985 564
"Bank of New York Computer Problem Disrupts Securities Marketplace," article from Investor's Daily, dated November 25, 1985 565
"Bank of New York Episode Catches Eyes of House Panel," article from the American Banker, dated December 9, 1985 569
"Computer Failure Disrupts Treasury Market," article from the American Banker, dated December 2, 1985 568
"Computer Snarled New York Bank," article from the Washington Post, dated December 13, 1985 571
"Corrigan, Bank of New York Chief to Testify In House Probe on Computer Failure," article from the American Banker, dated December 10, 1985 570
"FCS Can Borrow At Discount Window, Fed Testifies," article from the American Banker News Weekly, dated December 3, 1985 574
"Fed Considers New Penalty," article from the New York Times, dated December 13, 1985 575
Martin, Hon. Preston, Vice Chairman, Board of Governors of the Federal Reserve System, letter to Chairman Walter Fauntroy dated November 27, 1985, with attached letter from E. Gerald Corrigan, president of the Federal Reserve Bank of New York explaining the serious operational problems of The Bank of New York that occurred on November 21, 1985 247

Software development and security documentation: Page
Application System Development Controls (chapter 12) 377
Background Information on System Development from GAO's System Audit Training Material for GAO Evaluators 366
Computer Software Management: A Primer for Project Management and Quality Control (NSB Special Publication 500-11) 388
Guidelines for ADP Contingency Planning (FIPS Publication 87) 505
Guidelines for Automatic Data Processing Risk Analysis (FIPS Publication 65) 440
Guidelines for Security of Computer Applications (FIPS Publication 73) 467
NASA Guidelines for Assuring the Adequacy and Appropriateness of Security Safeguards in Sensitive Applications—Chapter 2: The Software Development Life Cycle and Security 273
Office of Management and Budget Circular A-71: Security of Federal Automated Information Systems 340
Overview of Computer Security Certification and Accreditation (NSB Special Publication 500-109) 346
The Complete Guide to Software Testing Chapter 7—Testing Systems 534
St Germain, Hon. Fernand J., Chairman, House Banking, Finance and Urban Affairs Committee, letter to Hon. Paul A. Volcker, Chairman, Board of Governors of the Federal Reserve System, dated December 3, 1985, requesting explanation of the Federal Reserve's loan to The Bank of New York in the amount of $22.6 billion advanced thru its Discount Window due to serious operational problem by the bank on November 21, 1985 252
The Bank of New York, document entitled "Government Securities Clearance Service" (GSCS) 209
"Volcker Downplays Loan to Bank of New York," article from the American Banker, dated December 13, 1985 573
Volcker, Hon. Paul A., letter dated January 23, 1986 with responses to questions from Chairman St Germain related to the Discount Window Loan to Bank of New York on November 21, 1985 254

THE FEDERAL RESERVE BANK OF NEW YORK DISCOUNT WINDOW ADVANCE OF $22.6 BILLION EXTENDED TO THE BANK OF NEW YORK

THURSDAY, DECEMBER 12, 1985

House of Representatives,
Subcommittee on Domestic Monetary Policy,
Committee on Banking, Finance and Urban Affairs,
Washington, DC.

The subcommittee met, pursuant to notice, at 9:30 a.m., in room 2222, Rayburn House Office Building, Hon. Walter E. Fauntroy (chairman of the subcommittee) presiding.

Present: Chairman Fauntroy; Representatives Carper, McCollum, Hiler, and Bartlett.

Chairman Fauntroy. The subcommittee will come to order.

On Thursday, November 21, 1985, the Federal Reserve Bank of New York extended a discount window loan of $22.6 billion to The Bank of New York. The loan was given to offset the accumulation of $20 billion in daylight overdrafts by The Bank of New York due to operational problems. These computer problems resulted in The Bank of New York being unable to deliver more than $20 billion in Government securities to their purchasers. The Federal Reserve was then compelled to step in to socialize both the potential losses and the embarrassment of a major clearing bank.

According to correspondence from the New York Federal Reserve Bank to the Board of Governors, The Bank of New York informed the Federal Reserve bank of the difficulties in early afternoon, yet issued continued assurances to the Federal Reserve bank that the problem would be corrected within a reasonable period of time. The letter went on to state that "these assurances were repeated frequently over the course of the day."

The Bank of New York, with what I must assume was the tacit agreement of the Federal Reserve, continued to accumulate an overdraft during the course of the day. Instead of requesting that dealers clear their transactions through other clearing banks, and thus creating a far smaller overdraft, The Bank of New York engaged in a high-risk strategy; namely, to conduct business as usual, assuming that the problem would be corrected by the end of the day. Clearly, it was not corrected, as the fourth quarter profit and loss statement for The Bank of New York will presumably attest.

This type of behavior, together with the size of the discount window loan, the largest loan ever made through the discount window by the Federal Reserve, raises grave concerns regarding the safety, soundness, and effectiveness of the payments system, the extent of daylight overdrafts and the Federal Reserve's oversight of the FEDWIRE system. What would have happened if the Federal Reserve had not stepped in to provide The Bank of New York with the $22.6 billion? I am profoundly, seriously, and painfully concerned that the implications might have been disastrous for the whole domestic and international financial system.

Nonetheless, the point remains that the Federal Reserve did take action. The action itself, however, raises additional concerns. Now that a clearing bank has experienced an overdraft 33 times the size of its capital base, and the lender of last resort has played its trump card by intervening on behalf of the bank, what is to stop other clearing banks from betting a similar strategy should they experience operational problems? And what about nonbanks? Will the Federal Reserve step in to provide a primary dealer with the needed liquidity? What about nonprimary dealers? How about the smaller banks and institutions that inhabit our land? Will the Federal Reserve step in to aid any large, troubled institution? Or are banks special? These questions must be addressed, and we must have answers so that we know the policy.

On a more specific level, we must learn about what happened at The Bank of New York on that fateful Thursday. So we address the following questions, as I have indicated to the witnesses.

First, when did The Bank of New York know about these difficulties?

Second, why did it not shut down once it did know?

Third, why did the Federal Reserve allow the bank to continue to do business and accumulate 20 times its capital position in overdrafts?

Fourth, why didn't the Federal Reserve Bank of New York pursue a more aggressive role in resolving and redirecting securities traffic away from The Bank of New York?

Fifth, how can the Federal Reserve better maintain control over daylight overdrafts?

Sixth, should the Federal Reserve continue to socialize the losses and embarrassments of participants in the market?

Seventh, should top-level management at The Bank of New York be removed immediately?

Eighth, should the bank have been considered insolvent?

Ninth, given the discount window loan from the Federal Reserve to The Bank of New York, what is to prevent other clearing banks from engaging in similar high-risk strategies in the future?

Tenth, how can the Federal Reserve better maintain control of the market?

Eleventh, should the Federal Reserve undertake to incorporate the operations of the clearing banks as a priced service?

In concluding my opening statement, let me quote from a publication by The Bank of New York concerning the Government Securities Clearance Service:

> The Bank of New York, with a reputation for innovative thinking and modern processing techniques, regards the Government Security Clearance Service as the ultimate blend of organization and technology.

The Bank of New York goes on to say to potential customers that:

> If increased profits are your goal * * * the GSCS system will reduce your clearance costs, alleviate transaction congestion and provide the ultimate in safety, security, and privacy. The primary benefit, however, is the virtual elimination of carrying costs resulting from unnecessary fails.

In spite of the assurances issued above, the system failed, whether unnecessarily or otherwise. What went wrong with the innovative thinking and modern processing techniques employed by The Bank of New York, and how do we prevent a recurrence by this or any other institution?

[The following notice of hearing and witness list have been submitted for inclusion in the record:]

WALTER E. FAUNTROY, DISTRICT OF COLUMBIA,
CHAIRMAN

STEPHEN L. NEAL, NORTH CAROLINA
DOUG BARNARD JR., GEORGIA
CARROLL HUBBARD JR., KENTUCKY
BUDDY ROEMER, LOUISIANA
JIM COOPER, TENNESSEE
THOMAS R. CARPER, DELAWARE
BEN ERDREICH, ALABAMA

H2-109 ANNEX NO. 2
WASHINGTON, DC 20515
(202) 226-7315

BILL McCOLLUM, FLORIDA
JOHN HILER, INDIANA
JIM LEACH, IOWA
STEVE BARTLETT, TEXAS
THOMAS J. RIDGE, PENNSYLVANIA
ROD CHANDLER, WASHINGTON

U.S. HOUSE OF REPRESENTATIVES
SUBCOMMITTEE ON DOMESTIC MONETARY POLICY
OF THE
COMMITTEE ON BANKING, FINANCE AND URBAN AFFAIRS
NINETY-NINTH CONGRESS
WASHINGTON, DC 20515

MEMORANDUM

December 10, 1985

REMINDER OF SUBCOMMITTEE HEARING AND NOTICE OF ROOM CHANGE

TO: Members, Subcommittee on Domestic Monetary Policy
Members, Full Committee on Banking, Finance and Urban Affairs

FROM: Walter E. Fauntroy
Chairman
Subcommittee on Domestic Monetary Policy

RE: Hearing on the Federal Reserve Bank of New York Discount Window Advance of $22.6 Billion extended to the Bank of New York

---

THURSDAY

DECEMBER 12, 1985

The Subcommittee on Domestic Monetary Policy will meet on THURSDAY, DECEMBER 12, 1985 at 9:30 A.M. in Room 2222 of the Rayburn House Office Building to receive testimony regarding the Discount Window advance of $22.6 Billion by the Federal Reserve Bank of New York to the Bank of New York.

9:30 A.M.

2222 RAYBURN HOB

In as much as the Full Committee is scheduled to mark-up two bills at 10:00 A.M., the Subcommittee will begin promptly and take such recesses as are necessary for Committee votes and other purposes.

WITNESSES:

The Honorable Paul A. Volcker
Chairman
Board of Governors of the Federal Reserve System

The Honorable E. Gerald Corrigan
President
Federal Reserve Bank of New York

Mr. J. Carter Bacot
Chairman and Chief Executive Officer
The Bank of New York

Chairman **Fauntroy**. To answer these and other questions, we will begin with our first witness, the Honorable Paul A. Volcker, Chairman of the Board of Governors of the Federal Reserve System, who is accompanied by the Honorable E. Gerald Corrigan, president of the Federal Reserve Bank of New York. We will also hear from J. Carter Bacot, chairman and chief executive officer of The Bank of New York.

Before commencing our testimony, I am pleased to yield to our distinguished ranking member, the gentleman from Florida, Mr. McCollum.

Mr. **McCollum**. Thank you, Mr. Chairman.

I want to welcome Chairman Volcker and our distinguished witnesses this morning. Certainly, we live in a computer age and an age in which, it is my understanding, we could not do the financial transactions, the sale of Government securities we undertake without these innovative computer technology operations.

What I am curious about are not only the answers to the chairman's questions but somewhat fundamentally whether this was simply a computer glitch we need to overcome in this day and age in order to be able to sell Government securities and transact our modern-day financial transactions, or is it more significant than that?

I understand no one was actually in this case harmed, no cost was to anyone except to The Bank of New York itself. However, I am wondering if corrective steps which may be taken by this institution are something that others in the industry can do, what we might expect in the future from similar type of problems that could arise in, I suppose, any major institution today having this high degree of computer technology, and just what the Federal Reserve has in mind that could help ameliorate these types of problems as we go down the line, if indeed they are problems rather than simply glitches.

I thank the chairman and look forward to the testimony.

Chairman **Fauntroy**. I thank the gentleman.

Mr. Volcker, we are pleased to have you before our subcommittee yet another time. We look forward to your testimony.

STATEMENT OF HON. PAUL A. VOLCKER, CHAIRMAN, BOARD OF GOVERNORS OF THE FEDERAL RESERVE SYSTEM

Mr. **Volcker**. Thank you, Mr. Chairman.

I was struck by Mr. McCollum's use of the word "glitch." I had that in my statement and decided it minimized this problem too much. I think the problem raises policy issues, and I have a brief statement I would like to read to review some of those issues that have been of concern to us for some time. My remarks will be relatively brief. As you noted, Mr. Corrigan, the president of the Federal Reserve Bank of New York, is here. He can address the more detailed questions and the specific facts.

The settlement problem which resulted in the $22.6 billion loan to The Bank of New York was caused by a computer system software failure. The effects in this instance were of unprecedented magnitude, measured by the amount of overnight loan. But the effects in terms of market performance and risk were well contained.

It is also true that more limited computer interruptions, either at private participants or at one of the Reserve banks, are not unusual. The impact is typically small, reflected only in temporary delays of minutes or hours in operations, or in final settlement for a day's work. This time, the interruption was much more prolonged, extending overnight. Consequently, potentially serious implications for the payments system and the securities markets were highlighted, although they were avoided in this instance. Since Mr. Corrigan will be reviewing in some detail the particular circumstances surrounding The Bank, of New York borrowing, I will simply turn to some of the policy issues.

Like it or not, computers and their software systems—with the possibility of mechanical or human failure—are an integral part of the payments mechanism. The scale and speed of transactions today permit no other approach. It is therefore appropriate to ask what type of backup systems—both hardware and software—and controls should be required of participants in the payments system, especially those with potentially large exposures measured relative to their assets, capital, and any other measures.

That is a question that must in the first instance be faced by each participant. Those participants, however, face intense competitive pressures to minimize costs and cash balances. As participants in and regulators of the payments system, the Federal Reserve has the unique responsibility to see to it that there is a countervailing pressure to provide protection against unacceptable risks for the system as a whole.

In approaching that question, the Federal Reserve has tried to identify and assess the risks facing participants in the payments and settlement mechanisms, how these risks interact, and what can be done to limit them in a cost effective way.

In that connection, Mr. Chairman, I should emphasize that my remarks are really directed toward payment systems in general and not just a clearing system for Treasury securities, which is at issue in this particular incident. Actually, the clearance system for Treasury securities, in some respects, has fewer risks because of the nature of its operation than some other elements in the payment system.

In any event, viewed in that broader perspective, for some years the Federal Reserve has been actively encouraging participants to adopt measures and policies to limit risk in payments and settlement systems, and we are reinforcing our own computer facilities, including backup systems.

After long discussions with other interested parties, the Federal Reserve Board earlier this year, in May, issued a general policy statement addressing certain problems in this area. That statement specifically called upon participants in private funds transfer systems—including the so-called CHIP's system, which handles some 100,000 individual, international, payments transactions, valued at several hundred billion dollars, per day—to better evaluate and control risks inherent in large-scale automated transfers.

We also announced at that time measures to control and reduce so-called daylight overdrafts on our own books—overdrafts which occur when, in the course of a day, a bank exhausts its reserve balance with a Federal Reserve bank.

In the last analysis, no mechanical system can be entirely fail-safe and also be commercially viable. The costs would simply be too high, and the money and Treasury securities markets could not operate at the present level of efficiency and low cost. Nor can key clearing operations be easily closed down in the middle of a day without potentially impacting severely on markets and third parties, sowing confusion at the least, and at worst a chain reaction of losses.

In these circumstances, the importance of institutions having access to the discount window is evident; in this instance, we could extend credit with the knowledge we were dealing with a known and reputable depository institution, supervised by Federal authorities. In fact, in this particular case, supervised by ourselves.

The discount window advance to The Bank of New York was, by any measure, enormous, but the collateral in our hands—U.S. Government securities that had been delivered to us for the account of The Bank of New York—was sound, and the Reserve bank also had further security from The Bank of New York. The market could, and did, proceed with its business, with minimal disruption.

In contrast, had the Federal Reserve Bank of New York refused to make payments on behalf of The Bank of New York as it received Government securities for its account, other market participants would have found themselves short of cash, other banks and their customers presumably would have been forced into overdraft, and requests for discount window assistance and financial pressures would have appeared elsewhere.

A question about the interest rate charged The Bank of New York for the use of the discount window in this circumstance is entirely appropriate. I have been assured—and Mr. Corrigan will explain more fully—that the net result of all financial transactions between the Federal Reserve and The Bank of New York was to offset fully the subsidy arising from the fact that the discount rate was below the Federal funds rate prevailing that day.

Those particular results were, however, fortuitous. At the same time, The Bank of New York did incur substantial expenses because it had to finance overnight some 25 billion dollars' worth of securities, upon which it received no interest. Notwithstanding that circumstance, a special penalty rate, designed to encourage better backup systems, when exceptionally large borrowing is caused by the institution's own computer problems, may well be appropriate. Over time, we will also be reviewing, as already contemplated, our policies toward tolerable levels of daylight overdrafts.

Your letter, Mr. Chairman, also asks whether the Federal Reserve itself should play a larger role directly in clearing securities—as a priced service—in order to reduce the overall risks to the system. That, frankly, is an area in which we would be extremely reluctant to enter, but we will be glad to provide further analysis of the advantages and disadvantages.

I believe it would be wrong to overdramatize this incident. There was a serious operational problem which illustrated some potential vulnerabilities in the clearing mechanism. But it is also true that the problem could be dealt with effectively within our present arrangements. In that sense, the system did, in this instance, prove fail-safe. The overnight loan, huge as it was, was fully secured, with an ample margin of protection.

But the incident is also indicative of the relevance of our continuing efforts—and that of the banks—to control risk in the payments system and of effective supervision of the participants. That work may seem mundane and tedious—that is, until something goes wrong. Then, it is also seen as essential.

[The prepared statement of Mr. Volcker follows:]

For release on delivery
9:30 A.M., E.S.T.
December 12, 1985

Statement by

Paul A. Volcker

Chairman, Board of Governors of the Federal Reserve System

before the

Subcommittee on Domestic Monetary Policy

of the

Committee on Banking, Finance and Urban Affairs

House of Representatives

December 12, 1985

I appreciate the opportunity to discuss with you questions relating to the operational problems experienced by the Bank of New York on November 21 and the response of the Federal Reserve Bank of New York. My remarks will be relatively brief. Mr. Corrigan, President of the New York Federal Reserve Bank, who was on the scene and here with you today, is in a position to review the specific facts and the Federal Reserve response to the events as they unfolded in full detail.

The settlement problem which resulted in the $22.6 billion loan to the Bank of New York was caused by a computer system software failure. The effects in this instance were of unprecedented magnitude, measured by the amount of the overnight loan. But the effects in terms of market performance and risk were well contained.

It is also true that more limited computer interruptions, either at private participants or at one of the

Reserve Banks, are not unusual. The impact is typically small, reflected only in temporary delays of minutes or hours in operations or in final settlement for a day's work. This time, the interruption was much more prolonged, extending overnight. Consequently, potentially serious implications for the payments system and the securities markets were highlighted although they were avoided in this instance.

Since Mr. Corrigan will be reviewing in some detail the particular circumstances surrounding the BONY borrowing, I will simply turn to some of the policy issues.

Like it or not, computers and their software systems -- with the possibility of mechanical or human failure -- are an integral part of the payments mechanism. The scale and speed of transactions permit no other approach. It is therefore appropriate to ask what type of backup systems -- both hardware and software -- and controls should be required of participants in the payments system, especially those with potentially large exposures measured relative to assets, capital, and any other measures.

That is a question that must in the first instance be faced by each participant. Those participants, however, also face intense competitive pressures to minimize costs and cash balances. As participants in and regulators of the payments system, the Federal Reserve has the responsibility to see to it that there is a countervailing pressure to provide protection against unacceptable risks for the system as a whole.

In approaching that question, the Federal Reserve has tried to identify and assess the risks facing participants in the payments and settlement mechanisms, how these risks interact, and what can be done to limit them in a cost-effective way.

For some years, the Federal Reserve has been actively encouraging participants to adopt measures and policies to limit risk in payments and settlement systems and we are reinforcing our own computer facilities, including back-up systems. After long discussions with other interested parties, the Federal Reserve Board earlier this year, in May, issued a policy statement addressing certain problems in this area. That statement called upon participants in private funds transfer systems -- including the so-called CHIP's system which handles some hundred thousand individual international payments transactions, valued at several hundred billion dollars, per day -- to better evaluate and control risks inherent in large scale automated transfers. We also announced at that time measures to control and reduce so-called "daylight overdrafts" on our own books -- overdrafts which occur when, in the course of a day, a bank exhausts its reserve balance with a Federal Reserve Bank.

In the last analysis, no mechanical system can be entirely "fail-safe" and also be commercially viable. The costs would simply be too high, and the money and Treasury securities markets could not operate at the present level of efficiency. Nor can key clearing operations be easily closed down in the middle of a day without potentially impacting severely on markets and third parties, sowing confusion at the least, and at worst a chain reaction of losses. In these circumstances, the importance of institutions having access to the discount window is evident; in this instance, we could extend credit with the knowledge that we were dealing with a known and reputable depository institution, supervised by federal authorities.

The discount window advance to the BONY was, by any measure, enormous, but the collateral in our hands -- U.S. Government securities that had been delivered to us for the account of BONY -- was sound and the Reserve Bank also had further security from BONY. The market could and did proceed with its business, with minimal disruption. In contrast, had the Federal Reserve Bank of New York refused to make payments on behalf of BONY as it received government securities for its account, other market participants would have found themselves short of cash, other banks and their customers presumably would have been forced into overdraft, and requests for discount window assistance, and financial pressures, would have appeared elsewhere.

A question about the interest rate charged BONY for the use of the discount window in this circumstance is entirely appropriate. I have been assured, and Mr. Corrigan will explain more fully, that the net result of all financial transactions between the Federal Reserve and BONY was to offset fully the "subsidy" arising from the fact that the discount rate was below the federal funds rate prevailing that day.

Those particular results were, however, fortuitous. At the same time, BONY did incur substantial expenses because it had to finance overnight some $25 billion of securities, upon which it received no interest. Notwithstanding that circumstance, a special penalty rate, designed to encourage better backup systems, when exceptionally large borrowing is caused by the institution's own computer problems may well be appropriate. Over time we will also be reviewing, as already contemplated, our policies toward tolerable levels of daylight overdrafts.

Your letter, Mr. Chairman, also asks whether the Federal Reserve itself should play a larger role directly in clearing securities -- as a priced service -- in order to reduce the overall risks to the System. That, frankly, is an area in which we would be extremely reluctant to enter, but we will be glad to provide further analysis of the advantages and disadvantages.

I believe it would be wrong to over-dramatize this incident. There was a serious operational problem which illustrated some potential vulnerabilities in the clearing mechanism. But it is also true that the problem could be dealt with effectively within our present arrangements -- in that sense the system did, in this instance, prove "fail-safe." The overnight loan, huge as it was, was fully secured, with an ample margin of protection.

But the incident is also indicative of the relevance of our continuing efforts -- and that of the banks -- to control risk in the payments system and of effective supervision of the participants. That work may seem mundane and tedious -- that is, until something goes wrong. Then, it is also seen as essential.

* * * * *

Chairman **Fauntroy**. Thank you, Mr. Volcker, for your statement.

Let me first say that I believe the Federal Reserve did take the proper actions with respect to the problem of The Bank of New York. While there may be questions about the rate, timing, and the type of examinations which we ought to be performing, I do want to make it clear that I believe the Federal Reserve, acting as this Nation's central bank, must continue to be prepared to undertake interventions which will stabilize and assure orderly markets.

Having said that, however, let me also make it clear that no one should expect the central bank to intervene to accommodate sloppy practices, without exacting a very high price.

Your suggestion that it may well be appropriate for special penalty rates which would be designed to encourage backup systems is well taken. Have you given much thought to what that penalty formula or rate might be?

Mr. **Volcker**. We have obviously begun to give it some thought, obviously, Mr. Chairman. We face a kind of policy or legal issue as to whether, if such penalties are appropriate, they should be through a special discount rate or whether the penalty might take another form.

We have penalties, as I am sure will come out in this hearing, for overdrafts that are extended overnight. That should not happen, ordinarily. We have rather severe penalties when that does happen. We might possibly adapt that kind of technique to this kind of situation, rather than do it by a special add-on, in effect, to the discount rate. We might do both. We will be reviewing these issues in terms of our authority under the law and otherwise.

I think the general problem we face here--and you have identified it very well, Mr. Chairman—is that, I think in the national interests, we want to provide, in the widest sense, backup, act as lender of last resort; but we do not want to do it in a way that gives incentives to the private market not to have incentives of its own and not to, in effect, bear the costs of operating these systems safely and effectively. So, providing penalties of this sort is one way of making sure that the costs fall on the operators of the System, or presumably enter into the charges paid by the people using the System.

Chairman **Fauntroy**. What do you think about the establishment of a clearing center that would permit institutions to net settlements to one another, thereby decreasing the number of transactions which must go over the FEDWIRE?

Mr. **Volcker**. We have some private systems springing up, funds transfer systems, that do provide an offset within themselves of payments to and from particular institutions. Then they come to the Federal Reserve for a so-called net settlement at the end of the day. Those systems have been of some concern to us, frankly, because they build up the equivalent of large overdrafts during the day among themselves, and if those overdrafts are not cleared up by the end of the day or are not manageable at the end of the day when they come to us, you have a mess.

That is one of the areas we have been concerned with. Not that those systems cannot fill a valid role, but they should be better operated with certain limitations on the amount of intraday credits that can be built up. That is one of the areas toward which we have been directing a lot of thinking. We have made certain policy decisions and then taken certain policy approaches to make sure those systems have within themselves safeguards that limit the credit risk involved in their own daylight overdrafts.

Chairman FAUNTROY. In my letter to you, I asked for copies of certain of the examination reports and procedures related to systems and of The Bank of New York. Were you able to get those to us?

Mr. VOLCKER. You might address that question to Mr. Corrigan. I hope we can work out something that can satisfy you without violating our general rule against giving out examination reports.

Let me say, in this connection, that we do review, as part of the examination process, these computer systems; but I do not think it is within our capability or anybody's capability to review them in the degree of detail that would have caught the kind of problem that arose here, which was basically a programming problem. I am sure the people that did that, and the bank itself, thought they had a program that would not be subject to this kind of a "glitch," and that they did a lot of testing. Nonetheless, these things sometimes arise.

We simply do not have the capability of assuring ourselves that every programming problem within these systems is caught or every contingency anticipated. We do look at them in a general way to see whether there are backup systems, what procedures they are following, the degree of management attention and control, and that kind of thing. And, we did do that for The Bank of New York.

Chairman FAUNTROY. As I indicated in my opening remarks, I am interested in the policy the Fed has, in terms of providing a lender of last sort function to a number of different financial entities.

I would like if you could, to state them for me, or at least give us an idea. What is your policy with respect to primary dealers who are not nonbanks?

Mr. VOLCKER. The only people we ordinarily would provide these facilities for are banks, depository institutions that by law have access to the Federal Reserve.

Chairman FAUNTROY. So, nonprimary dealers who are not banks would not, in your view, be eligible for this?

Mr. VOLCKER. Certainly not routinely. We have certain emergency provisions, but, no, they are not ordinarily eligible for discount lender assistance.

Chairman FAUNTROY. Would you say the same for nonbank institutions?

Mr. VOLCKER. Yes.

Chairman FAUNTROY. How about nonbank banks?

Mr. VOLCKER. The commercial parents of nonbank banks are not eligible.

Chairman FAUNTROY. Mr. McCollum.

Mr. MCCOLLUM. Thank you very much.

Chairman Volcker, is it true the United States couldn't sell the Government securities necessary to fund our deficits if we did not have the kind of high computer technology today in these primary dealers and other financial institutions?

Mr. **Volcker**. I think that is true as a general statement, yes.

Mr. **McCollum**. I gather from what you have just said, even though the chairman asked about punitive interest rates perhaps being charged by the Fed for sloppy practices, that you don't have a conclusion the particular case we are talking about here was the result of a sloppy practice. It was more of a glitch that might have been unavoidable, or is this a sloppy practice?

Mr. **Volcker**. I don't know that I can pronounce final judgment on that. I do know we have computer problems within the Federal Reserve, and other participants have computer problems from time to time, and other, very sophisticated computer systems, are subject to breakdown from time to time. Recognizing that as a fact of life, I think it is incumbent upon us and those that operate the system in the private sector to have as sophisticated, aggressive programs as they can to protect against these problems. Clearly, a question that arises here for the private banks and for us, frankly, is whether our backup facilities are both adequate and equipped to step in on short notice when the primary system goes down, as they are bound to do from time to time.

Mr. **McCollum**. Unfortunately, the only way to find that out is through tests like this?

Mr. **Volcker**. That certainly gives us certain insights into the problem. As I indicated, it is not unusual. Part of the operational problem facing the Federal Reserve Bank of New York was that it is not unusual to have a computer go down for a while; and usually they come up in a few minutes or an hour or so, and you can get through that period with some risks, but minimum risks, without any disruption to the market. What was unique about this situation was the computer never came back. By that time, they were deeply in the hole.

Mr. **McCollum**. I, for one, would not want to see the Fed get to the position of punitive interest charges to those institutions that have glitches that are of necessity part of the development of this process.

On the other hand, if there is a mechanism you developed so you can ultimately check whether they were careless or sloppy in the backup efforts to correct these problems, I would think that would be the standard you would be looking for.

I see you shaking your head.

Mr. **Volcker**. Certainly, the major thrust of our effort ought to be to use our supervisory and regulatory interest in the safety of the whole system to create, as I said in my statement, some countervailing pressure against the natural market pressures to do things as cheaply as possible. We are all interested in economy, but if the economy gives rise to excessive risks—which might be borne by third parties, in the last analysis—then I think we have a responsibility to make sure those systemic risks are dealt with by minimizing—we cannot eliminate them—the risks of this kind of breakdown.

Mr. **McCollum**. Mr. Chairman, I have other questions I could ask. I think I can ask them of Mr. Corrigan, as well. Perhaps the best thing is to let Mr. Bartlett ask his questions before we go vote.

Chairman **Fauntroy**. Mr. Bartlett.

Mr. **Bartlett**. Thank you, Mr. Chairman.

Mr. Volcker, the question I want to ask is, What do you think about the tax bill? But since I don't think you will answer it, let me ask something else.

In your judgment, then, is this situation largely an operational problem? Do you believe the Federal Reserve is taking steps to resolve the operational procedures to provide for the safety and soundness of the system? Or is there somehow more involved than that?

Mr. **Volcker**. This particular situation arose out of an operational problem. It very definitely illustrates a more general problem that is more than an operational problem. It raises the question of the amount of risks involved in these large-scale dollar transfer systems, whether it happens to be the Government securities clearing arrangements—where, as I indicated, the risks in some sense are less—or whether it is our own wire transfer system, the private wire transfer system, or the CHIP's system in New York that really does involve an extension of unsecured credit. To overstate it a bit, the extensions of credit during the day in the Government securities clearance system almost automatically provide their own security, because the Government securities are there when the cash is paid out. I may be overstating it a bit, but it comes close to that.

These other transfer systems simply have open overdrafts. The fact that you can have an extended computer breakdown of this sort highlights the fact that that is a real credit risk. Sometimes it is overlooked because the breakdown just happens during the day. That is something we have to be concerned about, and that is precisely the area in which we have devoted a great deal of attention over several years.

We have issued the policy statements that I referred to earlier. We have imposed certain approaches on banks and participants in the system so that they must review more carefully these credit risks they are taking during the day and explicitly set limits on them. We are still in the process of dealing with what is the appropriate policy toward limiting overdrafts for Government securities clearing mechanisms and transactions. So, it is an area that has occupied us.

I think we have engaged the attention of the major banks in this area, too. We have been working with them in, I believe, a cooperative way. Our perspectives, naturally, differ to some degree because there is this very obvious and legitimate urge to do things as cheaply as possible under the market pressures that exist. We have to balance that against the credit risks involved. That is precisely what we are trying to do.

Mr. **Bartlett**. It is your testimony, then, that the resolution of the problem really is in assessing and placing a credit limit, if you will, on daylight overdrafts?

Mr. **Volcker**. That is part of it. I do not know that that is the complete answer. That is certainly one area where we are directing much of our effort.

I might say this general type of problem is partly a chicken and egg problem. Everybody is trying to economize on cash. Banks are trying to economize on their use of reserves. Their customers are trying to economize on the use of cash in the banks. So that, looked at in another concentric circle, the banks have the problem with their own customers running overdrafts during the day. These all get pressed back on the central payment system. They are all, in one sense, a symptom of this effort and the technical capability of operating an economy on a very small cash balance and a very small reserve balance. I sometimes wonder whether the reserve balance and the basic cash in the system are enough, whether we are trying to run too many transactions on too narrow a cash base.

Chairman FAUNTROY. Thank you very much, Mr. Volcker.

I am still troubled about one thing which I cannot quite get clear on. Perhaps Mr. Corrigan or Mr. Bacot will also help me. You have indicated that no computerized payment system is fail-sale.

Mr. VOLCKER. I am no expert in this area, but computers seem to go down occasionally, and in that sense are not fail-safe.

Chairman FAUNTROY. That is what troubles me. I don't know if you have had the experience of standing in line for an airplane and the computer goes down. You lose time.

Mr. VOLCKER. Exactly.

Chairman FAUNTROY. We on the Capitol Hill from time to time have those down periods. It wreaks havoc on our access to information. If we cannot do an effective job, the cost efficiencies notwithstanding, at this level, with occurrences like happened on November 21, it thus troubles me greatly when we talk about going on star wars. Do you think we might tell the computer one thing, and it would drop a bomb on The Bank of New York?

Mr. VOLCKER. That thought has occurred to me. It is way outside my field of competence, Mr. Chairman. I assume they are not so constricted by costs and competitive pressures.

Chairman FAUNTROY. I am afraid it is outside mine as well, and that worries me—both the statement that nothing is fail-safe and the second one that may want to require some backup——

Mr. VOLCKER. To illustrate the problem, before this incident, going back some months, the Federal Reserve Bank of New York and all the Federal Reserve banks are involved, but New York has the biggest responsibility and operating load in this particular area, in the Federal Reserve System. They have been considering and have raised the question with us whether they shouldn't, in effect, duplicate their computer facilities as a backup in a different location so that it could be cut in immediately if there is a problem with their central computer. It is a good idea from one standpoint, but it also costs a lot of money. That is the balance that has to be made.

Chairman FAUNTROY. Thank you so much.

I will address more questions to Mr. Corrigan on that issue.

Mr. VOLCKER. I appreciate your attention, Mr. Chairman.

Chairman FAUNTROY. Thank you, Mr. Chairman. I know you have to move on to other things. We appreciate your coming.

Mr. Corrigan, it is a pleasure to have you here. We have looked forward to your testimony.

STATEMENT OF E. GERALD CORRIGAN, PRESIDENT, FEDERAL RESERVE BANK OF NEW YORK

Mr. CORRIGAN. Thank you, Mr. Chairman.

Good morning. I have to confess I have a little special chuckle out of your comment about star wars. I was chatting a bit with my colleagues last night. I was trying to think of a phrase that could capture the intricacies of the overall system that we are talking about. I thought of that one. They suggested maybe that wasn't a good idea for the reasons Chairman Volcker suggested.

Chairman FAUNTROY. But it certainly is thought provoking.

Mr. CORRIGAN. Mr. Chairman, in the interest of time I will read parts of this statement.

Two quick points to make at the beginning. The first point is on the operation of the book-entry clearance system itself. We have provided the subcommittee with a fairly detailed technical statement on that, which I will not go into at all.

The second point I would make is one already touched on. It is that when we look at the entire system of electronic payments, that involves thousands and thousands of entities, I think it is probably fair to say there are really only several dozen large participants, including the Reserve bank and most of the large commercial banks that constitute what I like to think of as the critical mass of the system, in the sense that a major interruption in processing by anyone of these entities does have the potential, at least, for more generalized problems elsewhere in the system.

Unfortunately, but inevitably, such disruptions do occur. For example, through early December of this year, there have been 70 days in which the closing hour of the securities wire has been extended by 2 hours or more. In rough orders of magnitude, about half of those extensions were due to what are called hardware problems and half software problems. Although in many instances, both hardware and software problems appear to have their roots in very large processing volumes on particular days. Looked at from a different vantage point, about half were due to problems in a Reserve bank and half in the very large commercial banks. However, of the 70 episodes, only 4 required the wire to remain open after 10 p.m. and, prior to November 21, none produced a situation in which the days' work was not ultimately completed.

Against that background, let me review, if I may, the events of that period on November 21 and 22. Early in the morning of the 21st, the Federal Reserve Bank of New York received a routine call from The Bank of New York indicating The Bank of New York was still in the process of completing its accounting and related work growing out of Wednesday's record volume of work and that they would not be in a position to begin Thursday's work until later in the morning.

In that timeframe, there was no indication of a major problem. However, between 8:15 and 10:15, all of the other major market participants began their normal work flows which entailed deliveries of securities to The Bank of New York—securities which in the normal course The Bank of New York would redeliver for payment to others. Because The Bank of New York was not in a position to process and redeliver these securities, the overdraft in The

Bank of New York's account began building the moment other banks commenced operations on Thursday morning.

At about 11:30 a.m. we first learned of a software problem at The Bank of New York. At that time, the overdraft in The Bank of New York's securities account with us had reached almost $12 billion. While the overdraft position was large for that time of day, it was not without precedent and was not cause for undue concern in a context in which there was still no indication of a sustained outage at The Bank of New York.

At about 2:15 p.m., a senior official at The Bank of New York called the Federal Reserve Bank of New York with the first firm indication that the problem at The Bank of New York might be much more serious than previously thought. At that point in time the overdraft in The Bank of New York's securities account was in excess of $20 billion.

Between 2:15 p.m., and 4 p.m., it became increasingly clear that the problem was of potentially major dimensions although prospects for recovering still seemed reasonable. However, it was also in this timeframe that it became very clear that The Bank of New York's data base of account information and individual transactions data had been damaged. As events were to prove, it was the damage to such records which severely complicated the recovery process and, as a practical matter, rendered unworkable several schemes which might have permitted the problem to be at least partially mitigated as the day progressed.

When at about 4 p.m. I personally learned of the apparent damage to data files at The Bank of New York, I immediately requested senior management at the Federal Reserve Bank of New York to establish contact with top management at The Bank of New York to get their appraisal of the situation. I also requested my associates at the Federal Reserve Bank of New York to begin drafting special loan agreements which could be used in the event that The Bank of New York could not solve their problem thus necessitating a large loan at the discount window.

Having been through a number of broadly similar situations, I remained hopeful that the problem would be solved but the fact of the damaged data files did concern me greatly. In this timeframe, The Bank of New York was indicating that it expected to be operational in the 6 to 7 p.m. period which, while late, would still have provided ample time to get the day's processing completed while permitting a comfortable margin of time for all parties to get their end of day accounting and related work completed for the next day's opening of business.

Between 4 p.m. and 8:30 p.m., there were numerous conversations between officials of the Federal Reserve Bank of New York and The Bank of New York and, at about 8 p.m., the special loan agreements were executed. At about 8:30 p.m., we were advised that The Bank of New York's data base problem had been solved and at 8:31 p.m., received the first securities transfer message from The Bank of New York of the day. At that point in time, The Bank of New York's overdraft was almost $30 billion.

By 10 p.m., we had seen only a trickle of securities traffic from The Bank of New York, and, it was in this timeframe that other problems and contraints had began to crowd in on the situation.

These included concerns about fatigue, technical constraints associated with keeping systems operational around the country after midnight and the need to close down the systems in time to permit all parties to do the end-of-day processsing to produce account statements for an orderly opening of business on Friday.

It is important to note here again, Mr. Chairman, that while problems somewhere in the system are not uncommon, we have never been confronted with a situation which required the system to be kept open past midnight. Nevertheless, the Federal Reserve Bank of New York decided to extend the processing hours as long as possible based on a judgment as to how close we could have it keeping in mind the time needed to do end-of-day processing.

Thus, at about 12:30 a.m., and confronted with increasing evidence of potential problems at other institutions around the country, the Fed broadcast a message nationwide that the securities wire would be closed at 1:30 a.m. and the funds wire at 2:15 a.m. The Bank of New York was notified of the decision in advance and was told to use the remaining time to process whatever it could and to go into the market and try to raise money in order to reduce further its overdraft.

However, given the hour, and because the operating system at The Bank of New York was, at best, limping along and because The Bank of New York quite naturally did not have such large credit lines established, The Bank of New York's debit position in its securities account was reduced only to $24.2 billion when the securities wire was closed at 1:30 a.m. After the close of the cash wire at 2:15 a.m., The Bank of New York requested and was granted a discount window advance of $22.6 billion.

As it turned out, The Bank of New York's estimate of its overall position at 2:15 a.m. was incorrect and, as a result, its reserve account was overdrawn by an additional $1 billion. Thus, the total credit extended to The Bank of New York by the Federal Reserve Bank of New York was $23.6 billion.

Given the special loan agreements that had been put in place and executed early Thursday evening, the decision to grant the discount window advance was an easy one since the only alternative at that time would have been an inadequately secured overdraft of $23.6 billion.

In that most fundamental sense, the important issues raised by this episode are not so much the fact of the discount window advance but rather its terms and, even more importantly, the underlying circumstances which permitted the build up in the overdraft in the reserve account in the first instance.

The absolute size of the discount window as noted earlier—is enormous by any standard. The loan is almost double the size of The Bank of New York and exceeds its capital by a factor of about $23 billion. It is far and away the largest overnight loan ever made through the discount window. This particular advance, however, should not be contrasted with large discount window advances to other institutions, such as extended credit provided to the Continental Illinois Bank, since the circumstances in these cases are fundamentally different. Under the terms of the special loan agreements, the loan was secured by all of The Bank of New York's domestic assets and by all of the customer securities they were empowered to pledge for such purposes. We estimate that the book value of the assets and securities available to secure the loan was about $36 billion thereby providing a very sizeable margin of excess collateral. Since the loan was adequately secured, the Federal Reserve Bank of New York was never subject to risk of loss and since the capital of The Bank of New York was not impaired its solvency was not in question.

The loan was made at the basic discount rate of 7.5 percent which was 54-basis points below the prevailing Federal funds rate that day. Thus, looking at the discount window advance itself, The Bank of New York's interest expenses were less than a "market" rate would have produced. However, when one takes account of all financial transactions between the Federal Reserve Bank of New York and The Bank of New York growing out of this episode those interest savings are more than offset. These transactions and other apsects of the situation are spelled out in greater detail in one of the appendices we have provided.

In brief, and specifically, because of the penalties assessed on The Bank of New York incident to the $1 billion overdraft in its reserve account and because The Bank of New York's problems prevented the delivery of certain securities to the Fed which The Bank of New York ended up financing but on which we, in effect, received the interest income, the net financial result of all transactions between The Bank of New York and the Fed was in favor of the Federal Reserve Bank of New York. But, this result was by chance and leaves very open, as mentioned earlier, the question of whether there should not be special terms associated with discount window lending of this nature.

As far as The Bank of New York is concerned, the overall episode entailed direct out-of-pocket costs of around $5 million which is about 7 percent of its earnings over the first three quarters of this year and, more importantly, is a sizeable percentage of the gross revenues it generates from its very clearing activities. The cost to The Bank of New York reflects the fact that it was required to finance almost $25 billion in securities overnight in a context in which the interest income on those securities accrued to those, including the Fed, who were due to receive the securities. Indeed, those individuals and institutions who bought the securities in question received a windfall in that they received interest for the day but did not incur any cost of financing.

Against this background, the subcommittee has raised a number of questions concerning the appropriateness of such a discount window advance being made at the basic discount rate and even larger questions as to whether, in the circumstances, the overall cost of the episode to The Bank of New York reflects the seriousness of the situation taken as a whole. With regard to the first of those questions, I believe the episode does point to the need for a reconsideration of the terms of discount window advances in cases like this.

The second question is, however, more difficult to deal with, for in the end it raises very difficult value judgments. Looked at narrowly, one can argue that the cost to The Bank of New York as referred to earlier is sizable, keeping in mind that there are indirect, as well as direct costs. On the other hand, it is also true that in the circumstances like prevailed on that day, The Bank of New York could not have funded itself any way other than at the Fed nor could it have repositioned or liquidated the securities in question because of the particular operating problem they had.

In the final analysis, therefore, the ultimate backup implicit in access to the Federal Reserve discount window supports the effective operation of the clearing system; and the question is whether those operating and paying for the System, in fact, bear its true costs, including the costs of adequate computer backup facilities. In this regard, I would hope that this experience, including its direct and indirect costs to The Bank of New York, would powerfully drive home the point that the oft-repeated concerns about payments risk are not simply a matter of abstraction but can be very real indeed and demand the attention of all who operate and manage the System including the top management of financial institutions.

Of course, The Bank of New York saga did not end with the extension of the discount window advance at 2:15 a.m. Friday morning. Indeed, as the normal business day opened later on Friday, The Bank of New York was still not operational. As a result by 11 a.m., it had accumulated a further overdraft in its reserve account of $2 billion. Faced with this situation, at about 11:30 a.m. we temporarily stopped accepting securities transfers for the account of The Bank of New York in an attempt to stabilize the situation somewhat and to see frankly whether it was practical to prevent further increases in the overdraft without causing excessive disruption in the market more generally.

As described in the next section, experience with this temporary stoppage illustrated all too vividly the potential for more generalized disorder that can result from such a partial shutdown of the market. Fortunately, shortly thereafter The Bank of New York regained full operational capabilities and with only a few further glitch, that word keeps coming up, Mr. McCollum.

Chairman **FAUNTROY**. Mr. Corrigan, the further you go with your testimony the less appropriate I think that term is.

Mr. **CORRIGAN**. I think it does understate things a bit.

We were able to complete the day's business and close the securities wire at 8 p.m. and the funds wire at 8:45 p.m. The Bank of New York had repaid its discount window advance in full and closed the day with a comfortable positive balance in its reserve account.

As noted earlier the real issues raised by this episode are not the discount window loan made at 2:15 a.m., Friday, except the question of the terms that are concerned. Rather the real issues are, first, whether there was some alternate course of action available during the day on Thursday which would have prevented the overdraft from developing in the first place and, two, if not, what can be done to mitigate against the reoccurrence of similar situations in the future.

Let me say a few words about the first of those questions at this time. Mr. Chairman, I have rigorously reconsidered, in retrospect, the question of whether on Thursday, November 21, we could have or should have done anything fundamentally different. My conclusion is that in the particular circumstances we faced, there was no reasonable alternative course of action available.

That is, given, first, experience with generally similar problems in the past; second, The Bank of New York's assurances that the problem would be fixed; third, the fact that The Bank of New York's overdraft was in excess of $15 billion as early as 12:30 p.m.; fourth, that the data files at The Bank of New York were damaged; fifth, that the solvency of the bank was not in question; sixth, that the opportunity to secure the loan was there, and seventh, perhaps most importantly, the enormous uncertainties, if not disruption that could result from a partial or complete shutdown of the market, I believe that our actions were prudent, disciplined, and appropriate.

In saying this, I should also confess that in some respects we were a bit lucky. The day in question was not the end of a reserve period settlement date; it was not a day in which there were other problems, technical or otherwise, in the markets; and unlike the preceding day, it was not a day in which the market was affecting settlement for billions of dollars in trading in mortgage-backed securities which had taken place over the prior month.

Let me also, if I may, Mr. Chairman make a few broader remarks about this situation. I think it is, in fact, very important that we step back from the particular episode here and look at the thing in a larger context. Much of the focus here today is on the Government securities clearance and settlement arrangement. That is, in part, only one large element of the overall payment system. As you know, we place considerable emphasis on our efforts to keep that overall system as efficient, reliable, trusted as possible. Again, much of the focus today is on a particularly vital element of that system, namely the clearance and settlement mechanism for Government securities.

Perhaps never has the need for an efficient market been more apparent than over the past month when, as a result of repeated delays in congressional passage of a permanent debt ceiling extension, the Treasury has had to announce plans on two occasions to auction for same day delivery and settlement new issues of Treasury bills, one of $22 billion and another of $15 billion all of which had to be sold, delivered, and settled within a period of hours.

On top of that, the Treasury was forced to compact into a period of 11 business days in November the auction and settlement of over $100 billion in bills, notes, and bonds. If I ever had any doubts, and I don't think I have, they were laid to rest on the second day of The Bank of New York clearance problem.

As I said earlier, the purpose of the temporary stoppage on securities transfers was to attempt to stabilize the situation and, frankly, also to see whether it was practical to prevent further increases in The Bank of New York's overdraft on our books without causing excessive market disruption. Operationally, this meant that holders of Government securities who had contracts to deliver those securities against payment to The Bank of New York for one of its customers were temporarily unable to make delivery under those contracts. And those firms which tried to send securities to The Bank of New York, including those whose computer systems were preprogrammed to make such deliveries, would have the transfers rejected back to them.

This temporary stoppage was only in effect for about 1½ hours, and it did not become generally known in the market until about 1 hour after it was in place. Yet, even in that short period of time, the result was a backup in the willingness and ability of some other market participants to transfer securities among themselves. This was especially true for other large clearing banks who were suddenly faced with the task of shutting off a segment of flows of preprogrammed, outgoing work in highly automated systems. They also found themselves in a position of potentially accumulating large overdrafts on our books since they could continue to receive transfers but were blocked from sending to a large value end point in the system.

Perhaps, most importantly, there was some evidence that investors were beginning to seek to break trades and financing transactions with dealers served by The Bank of New York. For those investors selling securities to such dealers or seeking financing from them, there was the sudden prospect that they could no longer count on making delivery against automatic payment, as well as the uncertainty as to whether they could be held financially responsible for a failure to make good on delivery.

Fortunately for all concerned, The Bank of New York was able to restore normal operations within a short time after the stoppage was put in place. So we did not have to confront a general stoppage in market activity or clearings that day.

Yet all of this did raise the specter of potentially larger and more disruptive problems, should for any reason market participants suddenly have to confront doubts about the ability to clear Government securities over the system in a safe, assured, low-risk fashion. So what I conclude from this experience, Mr. Chairman, is that, at least as things now stand and without potentially major changes in market practices and clearing techniques, it is unrealistic to think of disconnecting a major participant except in circumstances which, in the end, might and I emphasize might, require closing the market as a whole.

All of this naturally raises questions as to our current perspective on the broader issues related to payments system risk exposures. On that score, I think it is fair to say that the Federal Reserve has consistently been out front in calling attention to these issues and the need for initiatives to better control these risk exposures. Indeed, in the early going, our voice was something of a lonely one on these issues. More recently, the importance of these issues has been recognized by a broad cross section of bankers, supervisory authorities, and others with an interest in the strength of our financial system.

But even in the framework of this more constructive environment, the subject still has an aura of abstraction to it. Hopefully, the reality of a $22.6 billion loan will help to instill at least some religion into the remaining agnostics on payments system risk.

In looking to the future, Mr. Chairman, I think there are four points I would like to stress and then complete my comments as to the things I think should guide us as we try to look to a future in which we can have some greater assurances that we will not have these problems. The first is that the operation of the Nation's payment system is an intrinsic function of banks and the central bank and in my judgment is inexorably and irreversibly tied to the other functions of both. As an extension of this point, I am not at all sure that we can go on indefinitely processing an ever increasing volume of payments against an ever decreasing level of cash balances in the System.

Second, operating the payments system is not limited to pushing paper or computer blips, as important as that is, but fundamentally entails continuous extensions of credit and hundreds of credit decisions a day. To put it more graphically, the back office must be incorporated into the front office.

Third, all major payments system participants have public responsibilities including the responsibility to take that larger, longer look at the manner in which their decisions, their systems, and their behavior add to or detract from not just the speed and efficiency of the system but also its strength and integrity.

And, finally, since no system can ever be completely failsafe, in either a credit or operational sense, all major participants have a collective interest in pursuing initiatives to lower the aggregate risk in the system and to provide the means to continue operations despite a problem at any major link.

Now, my statement goes on to elaborate on those things. Let me go to the summary in terms of the things I think are technically important looking to the future.

In the course of the testimony, including the parts I have not read, I have referred to a number of steps I think warrant particular attention as we seek to provide a higher level of assurances that the episodes of the nature experienced on November 21 and 22 are not repeated.

In the very near term, the priorities seem to lie in four areas. First, considerations relating to the terms under which discount window credit is extended in similar-type situations; second, the development of better techniques to secure securities-related daylight overdrafts; third, efforts to explore possible opportunities whereby dealers and others may be able to bypass operational blockages even at a major institution; and, fourth, efforts to press ahead with stronger backup facilities on the part of the Fed and other major participants even though this may be quite costly.

In a somewhat larger timeframe, we must also look with care for opportunities to alter market practices and incentives in ways that can strengthen reliability and reduce risk while at the same time preserving the liquidity and efficiency of the market. That will not be easy, but it must be done.

[The prepared statement of Mr. Corrigan follows:]

For release on delivery
Expected at 9:30 A.M., E.S.T.
December 12, 1985

Statement of

E. Gerald Corrigan

President, Federal Reserve Bank of New York

before the

Subcommittee on Domestic Monetary Policy

of the

Committee on Banking, Finance, and Urban Affairs
U.S. House of Representatives

December 12, 1985

Mr. Chairman, members of the Subcommittee, I welcome this opportunity to appear before you to testify about the circumstances surrounding the malfunction in the government securities clearance mechanism which occurred on November 21-22. Before turning to the specifics of that episode, let me make a few introductory remarks about (1) the workings of the book-entry system for Treasury securities and (2) the structure and operation of the electronic network which permits the system to work with the speed and efficiency to which we are all so accustomed.

The most important operational characteristic of the book-entry system is its "payment against delivery" feature. That is, when bank A sells a security from its own account or from a customer account, the security is electronically delivered to bank B, and the reserve account of bank B is simultaneously and automatically debited to effect payment for the security. This feature of simultaneous and automatic payment in "final" funds is central to the efficiency and liquidity of the market. However, it is because of this feature of the system that it is common for large banks, especially large clearing banks, to have "daylight overdrafts" in their reserve accounts. And, it is because of this that the Federal Reserve has recently issued for public comment several proposals aimed at controlling daylight overdrafts growing out of such securities processing. These proposals, together with a more detailed discussion of the workings of the book-entry system, are summarized in Appendix A.

The electronic network over which transfers of securities and funds take place is a complex web involving the Federal Reserve Banks (and their branches), thousands of depository institutions throughout the country and thousands of customers of depository institutions. The electronic linkages between these entities take several forms ranging from the conventional telephone to direct interconnection by high-speed computer processors. While the network has thousands of participants, the vast, vast majority are not "critical" participants in that operational or other problem at most institutions do not have the potential for major consequences to others and can usually be worked around with relative ease. However, there are several dozen very large participants, certainly including the Reserve Banks and most of our large commercial banks, that constitute the "critical mass" of the system A major interruption in processing by any one of these entities does have the potential for more generalized problems elsewhere in the system.

Unfortunately, but inevitably, such disruptions do occur. For example, through December 5 of this year, there have been 70 days on which the closing of the securities wire has been extended by 2 hours or more. In rough orders of magnitude, about half of these extensions were due to hardware problems and half to software problems; although in many instances both hardware and software problems appear to have their roots in very large processing volumes on particular days. Looked at from a different vantage point, about half were due to problems in a Reserve Bank and half due to problems in the very large commercial banks. However, of the 70 episodes, only 4 required the wire to remain open after 10 p.m. and, prior to November 21, none produced a situation in which the day's work was not completed.

The Events of November 21-22

Against that background, the Federal Reserve Bank of New York received a routine phone call from Bank of New York at around 8:00 a.m., November 21, indicating that Bank of New York was still in the process of completing its accounting and related work growing out of Wednesday's record volume of work and that they would not be in a position to begin Thursday's securities processing until late in the morning. In this time frame there was no indication of a major problem at Bank of New York. However, between 8:15 a.m. and 10:15 a.m. all of the other major market participants began their normal work flows which entailed deliveries of securities to Bank of New York -- securities which in the normal course Bank of New York would redeliver for payment to others. Because Bank of New York was not in a position to process and redeliver these securities, the overdraft in Bank of New York's account began building the moment other banks commenced operations on Thursday morning.

At about 11:30 a.m. we first learned of a software problem at Bank of New York. At that time, the overdraft in Bank of New York's securities account had reached almost $ 12 billion. While the overdraft position was large for that time of day, it was not cause for undue concern in a context in which there was still no indication of a sustained outage at Bank of New York. At about 2:15 p.m., a senior official at Bank of New York called the Federal Reserve Bank of New York with the first firm indication that the problem at Bank of New York might be much more serious than previously thought. At this point in time the overdraft in Bank of New York's securities account was in excess of $20 billion.

Between 2:15 p.m. and 4:00 p.m., it became increasingly clear that the problem at Bank of New York was of potentially major dimensions although prospects for recovering still seemed reasonable. However, it was also in this time frame that it became very clear that Bank of New York's data base of account information and individual transactions data had been damaged. As events were to prove, it was the damage to such records which severely complicated the recovery process at Bank of New York and, as a practical matter, rendered unworkable several schemes which might have permitted the problem to be at least partially mitigated as the day proceeded.

When, at about 4:00 p.m. I personally learned of the apparent damage to data files at Bank of New York I immediately requested senior management at the Federal Reserve Bank of New York to establish contact with top management at Bank of New York to get their appraisal of the situation. I also requested my associates at the Federal Reserve Bank of New York to begin drafting special loan agreements which could be used in the event that Bank of New York could not solve their problem thus necessitating a large loan at the discount window. Having been through a number of broadly similar situations, I remained

hopeful that the problem would be solved but the fact of the damaged data files did concern me greatly. In this time frame, Bank of New York was indicating that it expected to be operational in the
6:00 p.m. to 7:00 p.m. period which, while late, would still provide ample time to get the day's processing completed while permitting a comfortable margin of time for all parties to get their end-of-day accounting and related work completed for the next day's opening of business.

Between 4:00 p.m. and 8:30 p.m. there were numerous conversations between officials of the Federal Reserve Bank of New York and Bank of New York and, at about 8:00 p.m., the special loan agreements were executed. At about 8:30 p.m., we were advised that Bank of New York's data base problem had been solved and at 8:31 p.m. received the first securities transfer message from Bank of New York of the day. At that point in time, Bank of New York's overdraft was almost $30 billion. By 10:00 p.m. we had seen only a trickle of securities transfers from Bank of New York. And, it was in this time frame that other problems and constraints began to crowd in on the situation. These included concerns about fatigue, technical constraints associated with keeping systems operational around the country after midnight and the need to close down the systems in time to permit all parties to do the end-of-day processing to produce account statements to permit an orderly opening of business on Friday. It is important to note here that while problems somewhere in the system are not uncommon, we have never been confronted with a situation which required the system to be kept open past midnight.

Nevertheless, the Federal Reserve Bank of New York decided to extend the processing hours as long as possible based on a judgment as to how close we could shave it keeping in mind the time needed to do end-of-day processing. Thus, at about 12:30 a.m., and confronted with increasing evidence of potential problems at other institutions around the country, the Federal Reserve Bank of New York broadcast a message nationwide that the securities wire would be closed at 1:30 a.m. and the funds wire would close at 2:15 a.m. Bank of New York was notified of the decision in advance and was told to use the remaining time to process whatever it could and also to go into the market and raise money in order to reduce further its overdraft. However, given the hour, and because the operating system at Bank of New York was, at best, limping along and because Bank of New York did not have such large credit lines established, Bank of New York's debit position in its securities account was reduced only to $24.2 billion when the securities wire was closed at 1:30 a.m. After the close of the cash wire at 2:15 a.m., Bank of New York requested and was granted a discount window advance of $22.6 billion.

As it turned out, Bank of New York's estimate of its overall position at 2:15 a.m. was incorrect and, as a result, its reserve account was overdrawn by an additional $1 billion. Thus, the total credit extended to Bank of New York by the Federal Reserve Bank of New York was $23.6 billion.

Given the special loan agreements that had been executed early Thursday evening, the decision to grant the discount window advance was an easy one since the only alternative at that time would have been an inadequately secured overdraft of $23 billion. In that most fundamental sense, the important issues raised by this episode are not so much the fact of the discount window advance but rather its terms and, even more importantly, the underlying circumstances which permitted the build up in the overdraft in the reserve account in the first instance.

The absolute size of the discount window advance is, of course, enormous by any standard. The loan is almost double the size of Bank of New York and exceeds its capital by a factor of about 23. It is far and away the largest overnight loan ever made through the discount window. This particularly advance, however, should not be contrasted with large discount window advances to other institutions, such as the extended credit provided to the Continental Illinois Bank, since the circumstances in these cases are fundamentally different.

Under the terms of the special loan agreements, the loan was secured by all of Bank of New York's domestic assets and by all of the customer securities they were empowered to pledge for such purposes. We estimate that the book value of the assets and securities available to secure the loan was about $36 billion thereby providing a sizeable margin of excess collateral. Since the loan was adequately secured, the Federal Reserve Bank of New York was never subject to risk of loss and since the capital of Bank of New York was not impaired its solvency was not in question.

The loan was made at the basic discount rate of 7.5 percent which was 54 basis points below the prevailing federal funds rate that day. Thus, looking at the discount window advance itself, Bank of New York's interest expenses were less than a "market" rate would have produced. However, when one takes accout of all the financial transactions between the Federal Reserve Bank of New York and Bank of New York growing out of this episode that interest savings is more than offset. These transactions and other aspects of the situation are spelled out in greater detail in Appendix B. Specifically, because of the penalties assessed on Bank of New York incident to the $1 billion overdraft in its reserve account and because Bank of New York's problems prevented the delivery of certain securities to the Federal Reserve Bank of New York which Bank of New York ended up financing but on which we, in effect, received the interest income, the net financial result of all transactions between Bank of New York and the Federal Reserve Bank of New York was in favor of the Federal Reserve Bank of New York. But, this result was by chance and leaves very open the question of whether there should not be special terms associated with discount window lending of the nature encountered in this case.

As far as Bank of New York is concerned, the overall episode entailed direct out-of-pocket costs of around $5 million which is about 7 percent of its earnings over the first three quarters of this year and, more importantly, is a sizeable percentage of the gross revenues it generates via its clearing activities. The cost to Bank of New York reflects the fact that it was required to finance almost $25 billion in securities overnight in a context in which the interest income on those securities accrued to those who were due to receive the securities. Indeed those individuals and institutions who bought the securities in question received a windfall in that they received interest for the day but did not incur any cost of financing.

Against this background, the Subcommittee has raised a number of questions concerning the appropriateness of such a discount window advance being made at the basic discount rate and even larger questions as to whether, in the circumstances, the overall cost of the episode to Bank of New York reflects the seriousness of the situation taken as a whole. With regard to the first of those questions, I believe the episode does point to the need for a reconsideration of the terms of discount window advances in situations like this.

The second question is, however, more difficult to deal with, for in the end it raises very difficult value judgments. Looked at narrowly, one can argue that the cost to Bank of New York as referred to earlier is sizeable, keeping in mind that there are indirect, as well as direct costs. On the other hand, it is also true that in the circumstances that prevailed on that day, Bank of New York could not have funded itself any way other than at the Fed nor could it have repositioned or liquidated the securities in question. In the final analysis, therefore, the ultimate back-up implicit in access to the Federal Reserve discount window supports the effective operation of the clearing system and the question is whether those operating and paying for the system in fact bear its true costs, including the cost of adequate computer back-up facilities. In this regard, I would hope that this experience--including its direct and indirect costs to Bank of

New York--would powerfully drive home the point that the oft-repeated concerns about payments risk are not simply a matter of abstraction but can be very real indeed and demand the attention of all who operate and manage the system including the top management of financial institutions.

Of course, the Bank of New York saga did not end with the extension of the discount window advance at 2:15 a.m. Friday. Indeed, as the normal business day opened later on Friday, Bank of New York still was not operational. As a result, by 11:00 a.m. it had accumulated a further overdraft in its reserve account of $2 billion. Faced with this situation, at about 11:30 a.m. we temporarily stopped accepting securities transfers for the account of Bank of New York in an attempt to stabilize the situation somewhat and to see whether it was practical to prevent further increases in the overdraft without causing excessive disruption in the market more generally. As described in the next section, experience with this temporary stoppage illustrated all too vividly the potential for more generalized disorder that can result from such a partial shutdown of the market.

Fortunately, shortly thereafter Bank of New York regained full operational capabilities and with only a few further glitches we were able to complete the day's business and close the securities wire at 8:00 p.m. and the funds wire at 8:45 p.m. Bank of New York had repaid its discount window advance in full and closed the day with a comfortable positive balance in its reserve account.

As noted earlier the most basic issues raised by this episode are not the discount window loan made at 2:15 a.m. Friday. Rather the real issues are (1) whether there was some alternate course of action available during the day on Thursday which would have prevented the overdraft from developing in the first place and (2) if not, what can be done to mitigate against the reoccurrence of a similar situation in the future. The remaining sections of my testimony deal with the second question but allow me to say a few words about the first at this time.

I have rigorously reconsidered, in retrospect, the question of whether on Thursday, November 21 we could have or should have done anything fundamentally different. My conclusion is that in the particular circumstances we faced there were no reasonable alternative course of action available. That is, given (1) experience with generally similar problems in the past; (2) Bank of New York's assurances that the problem would be fixed; (3) the fact that Bank of New York's overdraft was in excess of $15 billion by 12:30 p.m.; (4) that the data files at Bank of New York were damaged; (5) that the solvency of the bank was not in question; (6) that the opportunity to secure the loan was there; and, (7) the enormous uncertainties if not disruption that could result from a partial or complete shutdown of the market, I believe that our actions were prudent, disciplined and appropriate. In saying this, I should also confess that in some respects we were a bit lucky. The day in question was not and end of reserve period settlement date; it was not a day in which there were other problems--technical or otherwise--in the markets; and, unlike the preceding day, it was not a day in which the market was affecting settlement for billions of dollars in trading in mortgage-backed securities which had taken place over the prior month.

Importance of the Payments Mechanism

I think it is important to step back from the particular situation related to the Bank of New York's recent clearance problem and address some of the broader questions related to the operations of the payments mechanism. At the outset, let me stress again the importance we attach to maintaining an efficient and safe payments system. In a manner of speaking, it represents the financial equivalent of our interstate highway system, hooking together depository institutions of all sizes throughout the country, and through them financial and nonfinancial firms throughout the world. However, because of the value, volume and speed of traffic on the electronic version of the financial interstate highway system, the potential for disruptive chain reaction accidents is something we must take seriously.

Yet, there is a tendency to take for granted that the payments system will always do the job in a safe and efficient manner. In fact, it is only when there are highly visible disruptions to it that the public even takes some notice. And then it is typically only for a passing few days. So one of the reasons why we welcome these hearings, Mr. Chairman, is the opportunity it provides to reinforce the importance we attach to keeping our payments system efficient, reliable, and trusted.

Much of the focus today is on a particularly vital element of that overall system; namely, the clearance and settlement mechanisms for U.S. Government securities. As this Subcommittee well knows, the market for U.S. Government securities is the largest, the most efficient, and the most liquid securities market in the world. It is true that, in the first instance, the depth and resiliency of that market derives from the fact that the full faith and credit of the United States stands behind each U.S. Treasury security. But that market also draws considerable strength from the speed, efficiency, and low transactions costs associated with secondary market transfers over the book-entry system. These efficiencies contribute importantly to the liquidity of the market, to the attractiveness to investors of Treasury securities, to the Treasury's ability to carry out debt management operations, and to the New York's Fed's ability to carry out Federal Reserve System open market operations.

Perhaps never has the need for an efficient market been more apparent than over the past month when, as a result of repeated delays in Congressional passage of a permanent debt ceiling extension, the Treasury has had to:

- Announce plans on two occasions to auction for same day delivery and settlement new issues of Treasury bills, one of $22 billion and another of $15 billion all of which had to be sold, delivered and settled within a period of

> Compact into a period of 11 business days in November the auction and settlement of over $100 billion in bills, notes, and bonds.

It is difficult to imagine how debt management operations of this size would have been possible without the benefit of a highly efficient clearing and settlement system. Moreover, while these are somewhat unusual examples, just citing figures on the normal value and volume of Government securities which clear daily through this system--averaging about $200 billion and 27 thousand items--underscores the need for high speed efficient and reliable clearings. All participants in the market--including the dealers, investors, the Fed and the Treasury itself--have come to depend on this system. This dependence is not limited just to the efficiency and availability of the system. Rather, it extends in an ultimately crucial way to its safety and integrity as well.

If I ever had any doubts about this--and I don't think I have--they were laid to rest on the second day of the Bank of New York clearance problem. As I noted earlier, on that day we concluded it was necessary to establish a temporary stoppage on securities transfers being sent over the system to the Bank of New York. The purpose of this stoppage was to attempt to stabilize the situation somewhat and, frankly, also to see whether it was practical to prevent further increases in the Bank of New York's overdraft on our books without causing excessive market disruption. Operationally, this meant that holders of Government securities who had contracts to deliver those securities against payment to the Bank of New York for one of its customers were temporarily unable to make delivery under those contracts. And those firms which tried to send securities to the Bank of New York--including those whose computer systems were pre-programmed to make such deliveries once the securities were available--would have the transfers rejected back to them.

This temporary stoppage was only in effect for about one and a half hours and it did not become generally known in the market until about one hour after it was put in place. Yet, even in this short period of time, the result was a backup in the willingness and ability of some other market participants to transfer securities among themselves. This was especially true for other large clearing banks who were suddenly faced with the task of shutting off a segment of the flow of pre-programmed outgoing work in highly automated systems. They also found themselves in a position of potentially accumulating large overdrafts on our books since they could continue to receive transfers but were blocked from sending to a key large value end point. Perhaps most importantly, there was also some evidence that investors were beginning to seek to break trades and financing transactions with dealers serviced by the Bank of New York. For those investors selling securities to such dealers or seeking financing from them, there was the sudden prospect that they could no longer count on making delivery against automatic payment, as well as the uncertainty as to whether they would be held financially responsible for a failure to make good delivery.

Fortunately for all concerned, the Bank of New York was able to restore normal operations within a short time after the stoppage was put in place. So we did not have to confront a general stopage in market activity or clearings that day. Yet all this did raise the specter of potentially larger and more disruptive problems should for any reason market participants suddenly have to confront doubts about their ability to clear Government securities over this system in a safe, assured, low-risk fashion. So what I conclude from this experience is that, at least as things now exist and without potentially major changes in market practices and clearing techniques, it is unrealistic to think of "disconnecting" a major participant except in circumstances which, in the end, might require closing the market as a whole.

All this naturally raises questions as to our current perspective on the broader issues related to payments system risk exposures. On that score, I think it is fair to say that the Federal Reserve has consistently been out front in calling attention to these issues and the need for initiatives to better control these risk exposures. Indeed, in the early going, ours was something of a lonely voice on these issues. More recently, the importance of these issues has been recognized by a broad cross section of bankers, supervisory authorities, and others with an interest in the strength of our financial system. But, even in the framework of this more constructive environment, the subject still has an air of abstraction to it. Hopefully, the reality of a $22.6 billion loan will instill at least some religion into the remaining agnostics on payments system risk.

Against that background, allow me to suggest several broad elements of concern which, in my judgment, should remain central to our thinking as we seek to further enhance the reliability of the payments system. These points of emphasis are not new, but, if anything, our recent experiences only serves to strongly reinforce their value as a framework for addressing payments system risk issues:

First, operating the nation's payments system is an intrinsic function of banks and the central bank which is inexorably and irreversibly tied to the other functions of both. As an extension of this point, I'm not at all sure that we can go on indefinitely processing an ever increasing volume of payments against an ever decreasing level of cash balances in the system.

Second, operating the payments system is not limited to pushing paper or computer blips--as important as that is--but fundamentally entails continuous extensions of credit and hundreds of credit decisions a day. The "back office" must be incorporated into the front office.

Third, all major payments system participants have public responsibilities including the responsibility to take the larger, longer look at the manner in which their decisions, their systems, and their behavior add to or detract from not just the speed and efficiency of the system but also its strength and integrity. And, Fourth, since no system can ever be completely fail safe, in either a credit or operational sense, all major participants have a collective interest in pursuing initiatives to lower the aggregate risk in the system and to provide the means to continue operations despite a problem at any major link.

Let me expand briefly on each of these points and what they imply.

For some time now I have emphasized what I regard as the "special" functions performed by banks in our financial system and our economy generally--functions which justify the existence of the public safety net of Federal deposit insurance and central bank discount window support and which warrant the regulation and supervision of banks and companies which own and operate them. Operations of the payments system is one of those special core functions. From my perspective, this principle is firmly established in our traditions, if not laws, but is increasingly challenged by developments in the marketplace and, if I may be frank, by lack of developments in the Congress. The linkage between direct access to operations of the payments system, access to the banking safety net and acceptance of the regime of banking regulations and supervision is central to efforts to ensure the public interest in the strength and integrity of the payments system.

Just as I believe you can't separate operations of the payments system from the functions of banks and the central bank, I also feel that you can't separate large dollar payments processing and clearings from the credit extension process. The present operations of the book-entry clearing system amply demonstrate this point. As I have noted earlier, perhaps the single most important operational characteristic of that system is the ability to make delivery of securities against simultaneous receipt of final payment. This same feature often results in the clearing bank in effect extending daylight credit to its customers and in Reserve Banks extending such credit to the clearers. To be sure, that credit can be viewed in each case as supported by the underlying U.S. Government securities, but it still constitutes credit. Most other large dollar payment transactions do not have the built-in protections provided by the book-entry system as it pertains to payment against delivery of U.S. Government securities. Thus, if credit decisions are central to these transactions, they can only be even more important in the case of other large dollar payment transactions.

If, as I maintain, provision of large dollar payments and clearing services fundamentally entails managing credit exposures, then an important aspect of controlling payments risk is improved programs to address these credit exposures. For our part, we have been active on at least five fronts in this regard. First, we have forced a broad based dialogue on the general subject of payment risk and are now at the center of efforts to implement a program to control large dollar payments risk. Second, we are working closely with other bank supervisory agencies to promote strong internal bank policies and controls over settlement risk. Third, we are further stepping up our own bank examination efforts in the reviews of payments risk and clearing operations. In this regard, the

Board's recently announced initiative to expand the frequency and scope of examinations of large banks which will enable us to do some in-depth target examinations of clearing operations without foregoing a regular full scope safety and soundness review of the overall bank. Fourth, we are strengthening our internal systems and back-up capabilities on Fedwire to monitor developments and to control them as the need arises. And, fifth, we are expanding still further our analytical and empirical efforts to better understand the workings of financial markets and institutions so we can be better equipped to head off problems as well as to respond to problems if and when they arise.

As a general matter, bankers and their customers have also become much more attentive to the need to monitor and control credit exposure to payments risk. This heightened awareness to risk is a constructive development, but in a highly competitive environment awareness does not easily translate into concrete actions to limit risk across the board. And, at the other extreme, there is always the danger that reactions can go too far and lead to greater defensiveness or steps at self protection which can harm the functioning of the system as a whole. So even on this front we have to find the right balance between controlling credit exposures and not shutting off the flow of credit needed to keep the system functioning.

While much of our public emphasis and that of the industry has been on the credit risk issues, we have not been ignoring the need to strengthen the operational systems used for large dollar clearings and settlement. The case for effective back-up systems is clear but a vision as to what constitutes "effective" is somewhat elusive. A description of current back-up arrangements at the disposal of major participants is attached as Appendix C. At one time, "effective" implied that every major participant had redundant computer processors and communication lines, together with the ability to shift quickly from its main system to its back-up processor in the event of a problem. Then, when it became apparent that smooth uninterrupted electrical power couldn't be taken for granted, various forms of back-up power sources became more the rule than the exception. Still later, as the volume and value of transactions grew exponentially, finely tuned elements of contingency planning and exceptions management began to take hold in a context in which it was increasingly recognized that the implications of major and extended outages somewhere in the system--including at a Federal Reserve Bank--could be highly disruptive.

As impressive as these developments were, experience shows that more is needed. We must, for example, improve our ability to diagnose and respond to problems whether of the software variety or of some form of hardware or other physical disruption to a data communications center which could knock out both a main system and a back-up system, which are typically located in the same location. Within the Federal Reserve, we are developing remote site back-up arrangements at Culpeper, Virginia, to support our Fedwire system. But, in my personal view, even this may not be enough to assure sufficiently quick recovery from any major outage at the New York Reserve Bank. For this reason we have, for many months, been considering the costs and benefits of establishing a fully equipped back-up center within convenient reach of our main office. While such a facility would be very expensive, it seems to me that we must be prepared to give serious consideration to such an arrangement despite its costs.

Finally, while I have no doubt that more can and will be done by individual institutions to strengthen their operations and improve credit controls, it is unrealistic to expect that we will ever achieve a fail-safe payments system in either an operational or a credit sense. To use the Bank of New York episode as a case in point, it is not at all clear that even the most elaborate and expensive back-up system would have materially altered its capacity to respond to the particular problem once the apparent limitation in the software was in place since the same software would have been in place at a back-up facility. To me this underscores the collective interest that all participants in the payments and clearing systems have in exploring ways to limit the impact of operational disruptions or credit problems on the system. As far as the operational side is concerned, a common goal could be to have in place effective operating techniques to by-pass any one link in the system and thus permit the continued functioning of at least the largest payments despite an extended operational outage at any one participant. In terms of the credit side, this has to translate into a collective willingness to explore changes in clearing techniques and changes in market practices which would help reduce aggregate credit risk in the clearing system without impairing the smooth functioning of the markets which those systems serve. Neither of these goals will be easily achieved. On the operational side it will require further costly investments in back-up arrangments. On the credit side it means rethinking market practices which have evolved over many years and which should not be changed without careful consideration of possible unintended side effects.

III. Summary and Conclusions

In the course of the preceding elements of this testimony I have referred to a number of steps which seem to me to warrant particular attention as we seek to provide a higher level of assurance that episodes of the nature experienced on November 21-22 are not repeated. In the very near term, the priorities seem to lie in four areas: first, considerations relating to the terms under which discount window credit is extended in similar-type situations; second, the development of better techniques to secure securities-related daylight overdrafts; third, efforts to explore possible opportunities whereby dealers and others may be able to by-pass operational blockages even at a major institution; and, fourth, efforts to press ahead with stronger back-up facilities on the part of the Fed and other major participants even though this may be quite costly.

In a somewhat larger time frame, we must also look with care for opportunities to alter market practices and incentives in ways that can strengthen reliability and reduce risk while at the same time preserving the liquidity and efficiency of the market. That will not be easy, but it must be done.

Chairman **Fauntroy**. I thank you, Mr. Corrigan. I note for the record that you have been before this subcommittee before, on the question of examining the payment system. It has been a major concern of mine and the subcommittee for several years. In your testimony you raised the possibility of the establishment of a backup system for the New York Federal Reserve Bank. Let me make it clear I anticipate you will go forward with such a center. Frankly, I am concerned it has not already been done, particularly in light of previous testimony from you before the subcommittee on this very question. Beyond that position, which I think it is well known to the people in the system, let me ask you whether that system will have the capacity to process items—checks, securities, whatever—for other parties as well as for the Federal Reserve System itself?

Mr. **Corrigan**. Let me say a word first about this concept of what constitutes an effective level of backup. It is one that has evolved in its own way. The system has evolved and it has grown very big. There was a time where effective backup was thought of as a second computer. Then all of a sudden, we learned you could not count on electricity all the time, and then, a concept of backup meant having your own diesels and other noninterruptible power. Then it meant having an offsite facility, perhaps.

Even today, I think it should be stressed that many of these offsite facilities do not have the capacity to recover on the fly, so to put it. In other words, if a system goes down, for whatever reasons, in the middle of a business day, most of the backup systems that exist today, off premise, do not simultaneously record transactions data with the main system during the business day. Thus an institution cannot cut over from one system to the other. That is the feature that I am particularly interested in in respect to this idea that we have been pursuing at the Federal Reserve Bank of New York—not just a remote facility with another set of computers, but in addition to that, the capacity to, in effect, simultaneously log all transactions data at the main office and at the remote site. Thus if something goes down, you have the records, the transactions data, the accounts data at the remote site that you can get back in business in a matter of hours rather than in a matter of a day or more. That is an evolutionary concept in its own right. I think it is the way we have got to at least be prepared to think about it. In so far as the system as we are conceiving it is concerned, its main focus, would be the large dollar electronic payment systems.

We can handle the checks and other paper like that, one way or the other. But where the particualr problem arises, they are these large dollar systems, whether the Government securities or the Fed wire where you are moving hundreds of billions of dollars a day. If you are crippled in the middle of the day, recovery is very difficult.

Chairman **Fauntroy**. You mention in your testimony as well that this particular incident occurred at a very fortuitous time. What crisis management process you have in place to meet those kinds of needs, and what would you have done differently, for example, if, say, it occurred at the same time that ESM failed or a Continental Bank?

Mr. **Corrigan**. I like to say from time to time that we never manage crises, we just deal with little disruptions from time to time. One thing is very clear about all of these situations, Mr. Chairman. That is that they do vary case by case. There is no black book that can be invented or conceived in which you can anticipate in advance every little twist or wiggle that may arise in a particular set of circumstances. I think that my colleagues and I at the Federal Reserve bank and elsewhere in the Federal Reserve System do have a reasonably good sense as to how things work. That is one of the great values of the central bank having an operational role. We are in there; we do not have to start from scratch when these little disruptions arise.

There is no question that the delicacy of dealing with a problem like this gets a great deal amplified if you are working within a framework in which you have some other problem in the market. But I cannot give you a formula, because there is none there. I certainly will concede that it gets more difficult.

Chairman **Fauntroy**. I'm sure you are well aware of the concern the subcommittee has had concerning the payments systems. A recurring question has been the appropriate role for the Fed in clearing checks and other items. We dealt with that at a hearing last year. It concerns me that the Fed still seems reluctant to undertake a larger role in clearing securities, especially in light of the fact that a single clearer might dominate the market to an estimated 40 percent, thereby creating risks which could be minimized were the Fed more directly involved as, at least, one of the clearers. Why are you reluctant to take on this role?

Mr. **Corrigan**. I think in the most fundamental ways, it comes down to this difficult question as to where you perceive the legitimate role, the public sector versus the private sector. Traditionally, and I think with good reason, insofar as the payments mechanism is concerned, our role has been one that has been historically confined to direct relationships with banks. Once you have broadened that, say, to include direct working relationships, account relationships, and all the rest of it, with others, whether it is Government securities dealers, I think you cross a rubicon there in terms of what that says about the structure of the system.

If you are going to provide these services to Government securities dealers, does that mean they have access to the discount window? Do they have to have reserve requirements? It is, in a broad sense a very blood-and-guts public policy issue that opens up in that framework. My own continuing sense has been that our role is best conceived of one that essentially is a kind of linkage between the banks and one that does not reach out to non-bank entities. I feel rather strongly about it, because the minute you do it, you open up so many of these other questions. I think it is a private sector responsibility, plain and simple. However, if it turned out over some long period of time that the private sector could not build it, I obviously would have a different view.

Chairman **Fauntroy**. You are not there now?

Mr. **Corrigan**. No, sir.

Chairman **Fauntroy**. Mr. McCollum.

Mr. **McCollum**. Thank you, Mr. Chairman.

Mr. Corrigan, in Mr. Bacot's testimony that I know I have the advantage of having—I always wonder whether we should have the others first or whatever—he addresses a couple of our questions about what should be or should not be done. One has to do with something the chairman was alluding to a minute ago in the terms of the number of clearing banks and this sort of thing.

It suggests it would not result in any increased efficiency to have an increased number of them. In fact, it could compound the potential computer error. Do you think there should be more clearing banks? Do you think if there were more, or they were required, in this day and age we would compound the number of potential errors?

Mr. **Corrigan**. That is a tough question. Normally any aggressive proponent of free markets and competitive forces like myself would answer that by saying sure. But it is not that simple, because of a couple of possible problems. First of all, I do think there is at least something to the argument that says if you have 10 players instead of 4, have you increased or decreased the risks that mechanical snafus and other problems are going to happen either more often or less often. I am not sure which it would be.

But on top of that, I think an important question is: How much competition, to put it very bluntly, is good here? As it is right now, it seems to me there is a question as to whether enough capital in the form of backup equipment and so on is being devoted to this whole undertaking. There is at least a question as to whether those that provide these services are being compensated in a way that takes account not only of the direct costs of providing the service but also the risks that they may be incurring.

If I am correct on those things, is a larger number of clearing banks going to aggravate or ameliorate those concerns? I don't know. It is a tough issue. I don't have a preconceived view that there is some magic number. But I do not think it is simply a question of x is better than y.

Mr. **McCollum**. If you had a requirement that the computer software systems and computers be somewhat uniform or operated like the U.S. House does and only lets us buy certain very narrow ranges of products, you would be impeding free market innovative system that has really allowed the process of handling all of these huge numbers to keep ahead of the curve.

Mr. **Corrigan**. That is correct.

Mr. **McCollum**. I think there is a tremendous problem there, although it seems to me there is a problem for any backup system you might be developing. In this particular case, we are dealing with a software program failure; and, as I understand it, The Bank of New York software and program backup systems they have are unique.

How would you go about backing this up if you wanted to?

Mr. **Corrigan**. I think two points are relevant. In a part of my testimony that I did not read, I made the point in this context of backups—and, again, I am no technical expert here; let me not mislead you—that it was not clear to me that even the sophisticated kind of backup system that I described to the chairman a moment ago would have necessarily been able to overcome this particular problem. This is because, by definition, you would have had the same software in the backup facility as you had in the primary facility, even if that backup facility was the solid gold Rolls-Royce variety. I think we need to keep that in mind in the context of this question about what is fail-safe and what is not. But I do not think we are completely hamstrung, either. I am at least intrigued—and I want to emphasize that word "intrigued"—by the possibilities that we may be able to conceive of a system that might provide backup some other way. For example, The bank of New York has all these dealer customers out there, and they have computer-to-computer links or terminal links to The Bank of New York system.

The problem, of course, is that everything is preprogrammed. That is why you cannot divert it someplace else. But I think we ought to be looking at, and we intend to look at, the question of whether there should not be at least a limited capability at each major participant through something like a dial-up terminal, not a computer-to-computer link—to bypass a major troubled institution. In other words, dealer A's primary business relationship would still be with The Bank of New York. But there could be an arrangement whereby there is at least a limited capability, if The Bank of New York is down, to bypass it by going through another clearer, at least on an emergency basis. Those types of arrangements do not entail having to have standard computers and software up and down the line.

Mr. **McCollum**. The whole thing is intriguing. I think we have a lot to learn. We are learning it.

I asked the question of Mr. Volcker a while ago. I think, obviously, this is a learning experience itself.

I have one last question I want to ask about it. It is also brought up by something I read in Mr. Bacot's anticipated testimony dealing with a scheduled closing hour, and the fact there are direct delays in closing. He suggests that the problem with those delays in closing is not the scheduled closing hour but, rather, the need to distribute the clearing traffic more evenly throughout the day.

What is your assessment? Do you agree with that?

Mr. **Corrigan**. Let me make two points. The first was a question raised by the chairman's letter: Why don't we simply "solve that problem" by permanently extending the closeout? I think you anticipate the answer to that. My fear there, of course, is that if we extend the closeout on a regular basis, to 4, 5, 6 o'clock, and do nothing else, the peak will just move with the extension; and then when we have problems, we have less time left to try to solve them. Thus I think the early closeout, at least until we can get our arms around some of these other problems, is a form of insurance and a form of discipline.

But I think one of Mr. Bacot's comment in his testimony has considerable merit to it. I, for one, am not at all persuaded that there are not opportunities to alter market practices at least somewhat to try to spread out the volume load more evenly, even in the framework of a 2:30 closeout hour. I think, just as an example, we are going to be having a meeting in New York next week with all the clearing banks and many of the dealers to begin more aggressively raising questions of that very nature.

Mr. **McCollum**. Thank you very much, Mr. Corrigan.

Thank you, Mr. Chairman.

Chairman **Fauntroy**. Yes.

Mr. Hiler.

Mr. **Hiler**. I thank the chairman for yielding.

Mr. Corrigan, I have one or maybe two questions. It seems to me that the biggest danger that comes from the kind of problem that The Bank of New York had is that for an instant it creates an amount of uncertainty in the marketplace or at the Fed.

Mr. **Corrigan**. That is correct.

Mr. **Hiler**. There was no insolvency here, no lack of capital, any of those kinds of things. It was a glitch, a computer problem.

Do you feel reasonably confident in the New York Fed's ability to ascertain in fairly short order whether or not a problem that is being faced is, in fact, a technical problem or whether it is an underlying financial problem?

Mr. **Corrigan**. I think that our intelligence network on these kinds of questions is quite good. I would never take the position that it is as good as it could be, that is not in my nature.

I think that this is due, in part, to the attitudes, if I may say, of the banks and other major market participants—both domestic and foreign—in the New York market. By and large, they are pretty good in terms of keeping us informed and talking to us. So I think our ability to react quickly to a problem is good; good enough now. Are we trying to make it better—certainly.

Mr. **Hiler**. Do you have within the Federal Reserve bank a group of individuals who, when something like this occurs, kind of go into action to make that relatively quick decision so that you know whether you have a big problem on your hands or just a small problem?

Mr. **Corrigan**. This morning we talked about computer software, hardware, overdrafts, billions, all the rest of it. But the name of the game in all of this is still people. I have to say that when I look at the way our people responded on this—and I am not just talking about executives, I am talking about secretaries, clerks, computer analysts, lawyers, economists, as well as executives—I am not saying it is perfect, but I have to say I happen to be proud of them.

Mr. **Hiler**. Mr. Chairman, with that, I yield back.

Chairman **Fauntroy**. Thank you.

Mr. Corrigan, how dependent is the Government securities market on automated information systems? You sent a letter to Vice Chairman Martin (see appendix p. 249) indicating that, while the situation is obviously serious, it happened that similar types of problems have happened on numerous occasions. You reiterated today what you said to Mr. Martin; namely, that there have only been 2 days in the last year when you closed on time.

Mr. **Corrigan**. If you go back to two illustrative points I made in my testimony about what operations the Treasury has been able to do in the market in this past month—on two occasions, selling $15 billion and $22 billion in Treasury bills in a matter of hours; and, in 11 business days selling $100 billion in new securities; that is staggering. There is no way, in my judgment, that the market could even begin to function with the load that is being put on it were it not for these highly elaborate systems, systems that, because they are so elaborate, because they are sophisticated, because they are so efficient, are inherently subject to—I have to stop using the word "glitch" here—are subject to little disruptions here and there. But I do not see how you could have it any other way.

I think the emphasis has to be on, as I suggested in my testimony, trying to make the system that much better. But I do not think there is an alternative to the framework we are working in.

Chairman **Fauntroy**. What are you doing on that question right now?

Mr. **Corrigan**. Chairman Volcker mentioned, and I laid out four or five areas in my own testimony that I did not read. There are a number of initiatives that have been underway for some time on this whole question of daylight overdraft control exposure. We have our own initiatives in the areas of backup systems. The banks have the same type of initiatives.

We, at the New York Fed, have been involved, quite apart and before The Bank of New York problem, with a very aggressive diagnostic study to try to get beneath the symptoms that we see. We have, for example, a group of people that are taking a look at four or five segments of the market in a very aggressive way to try to understand the plumbing of the market, as I understand it, and look at that in the context of what we see in these large dollar payment systems. We have in mind—with the full cooperation, I might add, of the banks in New York—picking a day some time early next year that we know is going to be a very heavy volume day and, through various sampling procedures and whatever, trying to get underneath the flows and understand in minute detail what these transactions really represent, so we can have a better analytical understanding as to how the thing really fits together, where the weak spots are, and indeed where the circuit breakers are.

We have also been taking some initiatives on the bank supervisory side. For example, as you I am sure know, Mr. Chairman, the Federal Reserve Board announced 6 weeks or so ago a new schedule for examinations of the major banking institutions starting in 1986 wherein each year we plan to do—for the organizations that are under our jurisdiction, of course—a full-scale standard commercial examination.

But, in addition to that—not as a substitute for but in addition to—what is called in the trade a targeted examination that picks a particular facet of a bank's operation and go into that in considerable detail. Because we now have that program in place, one of the things that we will do as part of that program is perform at least one, and maybe more than one, targeted examination, in a very thorough way, of these clearing activities in major banks. Thus, there are a multiplicity of initiatives, all of which predate this particular problem.

Chairman **Fauntroy**. I wonder if you could just respond to two things quickly. Could you provide for the record background information on the degree to which the automated systems are being used to operate and manage the Treasury securities market, and how they interface with one another?

Mr. **Corrigan**. Sure.

[In response to the request of Chairman Fauntroy, the following information was received from Mr. Corrigan:]

REPLY RECEIVED FROM MR. CORRIGAN

Q. Could you provide for the record background information on the degree to which automated systems are being used to operate and manage the Treasury securities market? And how they interface with one another?

A. In an operational sense, the smooth functioning and liquidity of the market for Treasury securities is entirely dependent on automated systems. Nearly all Treasury securities are now issued in book-entry form and the Treasury plans to complete the conversion to full book-entry by mid-1986 with the implementation of its Treasury Direct Access Book-Entry Securities System. Following implementation of this system, no new issues of bonds and notes will be available in paper form, thereby completing the Treasury's goal of issuing all marketable securities in electronic form.

As for the secondary market activity, virtually all transactions in Treasury securities are conducted in electronic form over Fedwire, although some are effectively netted during processing in the automated systems at clearing banks and brokers. Major participants in the market, both dealers and investors, domestic and foreign, effect delivery and settlement of transactions solely by use of the automated telecommunications network--the Fedwire and the commercial book-entry custody system. In all cases, the clearing and book-entry systems at the large clearing banks are connected to the Fedwire system by high speed computer-to-computer linkages.

The degree of automation linkages among participants in the market, however, is not uniform. For primary dealers, the amount of automation runs the gamut from highly sophisticated direct links with their clearing banks to manual delivery of trade tickets for further processing by the clearing banks. This would be true of other major investors as well. However, no matter what the degree of automation that exists between a customer and its clearing bank, whether dealer or ultimate investor, the final transfer of Treasury securities must take place in an automated environment.

The extent of disruption to the market that can occur when any link in this chain of automation falters will depend on the volume of activity processed by the entity having problems and the length of the outage. A computer outage at a small dealer or investor will cause little disruption to the process while extended problems at a large clearing bank, which handles dozens of accounts and a large volume of transactions, will eventually have a major impact. In the worse case scenerio, a lengthy computer outage at a Reserve Bank, particularly the FRBNY, has the potential of temporarily closing the market.

Finally, it should be noted that market dependence on automation is not only limited to custody accounts, clearance and settlement operations. Rather, it also extends to trading, investment and financing operations as well. All major dealers and investors are dependent to some degree upon automated data systems to provide virtually instant access to the latest information on prices and news developments that might impact prices. Those with active trading and financing operations are further dependent on automated systems to monitor intraday positions and to support pricing decisions of complex financial transactions. However, while highly sophisticated systems at this level can add to the liquidity and efficiency of the market, they are not as critical as the clearing mechanism. Market participants have shown they are able to adapt operations quickly to deal with disruptions to internal and external information systems.

Chairman FAUNTROY. In 1984 a study of risks in the U.S. payment system was done on the CHIP's network. I understand that it simulated the various assumptions that might result in a crisis.

Has the Fed done that type of risk simulation? If so, would you provide it for us?

Mr. CORRIGAN. Yes, we have. Yes, we will.

[In response to the request of Chairman Fauntroy, the following information was received from Mr. Corrigan:]

REPLY RECEIVED FROM MR. CORRIGAN

Q. **Back in 1984 a study of risk in the U.S. payment system was done on CHIPS, the CHIPS network, that sort of simulated the various assumptions that might result in a crisis. I wonder if the Fed has done that type of risk simulation? If so, would you provide it for us?**

A. Staff in the Federal Reserve have conducted simulations of the potential effects on participants in the payments system of the sudden inability of a participant in a net settlement system (such as CHIPS) to settle its end of day position. By nature, these types of simulations cannot be expected to capture the many reactions of other participants both to the emergence of signs of a potential settlemenmnt failure, as well as to an indication of an actual problem. Some of those reactions can work to lessen the problems of a possible failure, while other types of reactions could exacerbate those risks. Instead, these types of simulations are typically performed on the basis of various static assumptions about the responses of other settlement participants. Thus, the value of this type of exercise is not as a tool to predict actual results of a settlement failure, but rather as a technique to highlight and roughly measure the nature and size of the interactive problems that might arise in such a situation.

Staff at the Board and the Federal Reserve Bank of Minneapolis performed such a simulation using data provided by CHIPS for actual transactions data for two different business days in January 1983. The results of that simulation of a hypothetical settlement failure on CHIPS were presented in a recent paper (May 13, 1985) published by David B. Humphrey, Assistant Director of the Division of Research and Statistics at the Board of Governors. The attached excerpt and tables from Mr. Humphrey's paper describe in detail the nature and results of that simulation.

EXCERPT FROM PAPER ENTITLED:
PAYMENTS FINALITY AND RISK OF SETTLEMENT FAILURE: IMPLICATIONS FOR FINANCIAL MARKETS

(PAGES 6 - 15)

BY

DAVID B. HUMPHREY
ASSOCIATE DIRECTOR
DIVISION OF RESEARCH AND STATISTICS
BOARD OF GOVERNORS OF THE FEDERAL RESERVE SYSTEM
WASHINGTON, D.C.

III. The Effects of a Settlement Failure: A Simulation Analysis.[5]

Possible Responses to a Settlement Failure. Pursuant to the CHIPS rules, the inability of a participant to settle its end-of-day net debit position could result in all its transfers sent and received being backed out of the settlement (a process called "unwinding"). A revised settlement position for the remaining participants is then calculated. In the real world, such an event would occur only after an intensive effort to work out a solution that would allow settlement of all activity without unwinding. Removing the transfers sent and received by a participant would have a different impact on different participants, depending on their net position with the institution that fails to settle. If an institution sent exactly the same aggregate dollar value of transfers to the failed participant as it received from the failed participant, it would have a net position of zero and its net position in the settlement would be unaltered. To the extent that an institution received a larger aggregate dollar value of transfers from the failed participant than it sent to the failed participant, it would have a net credit balance with that institution,

BEST COPY AVAILABLE

and the removal of all transfers between them would reduce the remaining institution's net credit position in the settlement (or increase its net debit position). Conversely, failure of an institution with which a bank has a net debit position will reduce the bank's net debit position in the settlement or increase its net credit position.

Because the removal of one institution's payments activity changes the settlement positions of other participants, there is the risk that the failure of one institution can precipitate the failure of others (systemic risk). Systemic risk occurs with a failure to settle on CHIPS (or CHESS), which is a provisional funds transfer network, but not FedWire (or CashWire).[6] A sender's failure to extinguish its daylight overdraft on FedWire creates credit risk for a Reserve Bank. Under Regulation J, Reserve Banks provide good and final funds to receivers of FedWire payments. Thus the failure to "settle" on FedWire can not affect other bank participants on the network, although the Reserve Bank can suffer a financial loss.

A number of factors will affect the degree of systemic risk associated with the default of a participant on its payment obligations on a provisional wire transfer network such as CHIPS. First, the sources of the shock that causes the bank's default will determine the context in which the default occurs. The more sudden and/or unexpected the shock and the resulting default, the less opportunity other institutions will have to reduce their exposure to the failed institution. Second, the number of institutions affected by a shock, the size of the institution(s) that defaults, and the magnitude of the net positions other institutions have with the defaulting institution(s) will all shape the outcome.

The options available to deal with the shock also bear on its consequences. There are several considerations that will determine the impact of any failure to settle. First, an institution's exposure to the removal from the settlement of another institution's funds transfers to it will depend on the use, if any, it has made or permitted customers to make of the provisional entries representing the funds transferred prior to settlement. With respect to messages received for customers' accounts, the extent to which the bank has permitted the customer to use these balances and the customer's obligation and ability to return any portion of the balance used will determine the impact of the nonsettlement of the transfer. If the customer has not used the funds represented by the transfer or can and does return them, the bank will not be harmed by the nonsettlement of the transfer.[7] Similarly, with respect to funds transfers received for its own account, to the extent that the bank has not relied on them in some fashion, their nonsettlement will not harm it.

If a bank is in a net debit position in the revised settlement, the nonsettlement of funds transferred for which it cannot readily reverse the use it has made or permitted of the funds may present it with a liquidity problem. The resulting need for additional funds to satisfy its revised settlement obligation may be met in one of several ways. Up to some point, an institution will be able to handle the revised settlement obligation from its own resources. For institutions with less than $250 million of capital, the "comfort letter" required by CHIPS also obligates the parent to use its resources to meet the obligations of the subsidiary (although an end-of-day failure may preclude any meaningful exercise of this option). Access to the discount window may, for certain institutions, also provide necessary funds. Finally, it must be remembered that CHIPS is a net-net settlement system with the activity of the participants being netted with that of their settling participants and settlement occurring first among the settling participants. A settling participant may be willing to settle for one of its associated participants' facing a liquidity problem in meeting its revised settlement obligation. All of these factors will determine a participant's ability to meet its revised settlement obligation.

The magnitude of the net position changes which would result from the sudden and unexpected failure of a CHIPS participant can be examined by taking the record of transaction activity for a given day and removing all transfers to and from a particular institution. Using such historical data, it is necessary to assume a precipitous failure of an institution, since there is no way to fully simulate other institutions perceiving a problem and partially adjusting their exposure. However, this problem can be at least partly remedied by removing the transactions to and from a participant in a net credit position for the day studied. The failure of a participant in a net credit position means that other participants have extended the failed institution less gross daylight credit than if it were in a large net debit position during the day and, in this sense, is similar to the likely response by network receivers to a questionable participant prior to its relatively unexpected failure to settle.

A Simulation Exercise: The Settlement Failure of a Large Participant.

Using the actual record of all transactions for a randomly selected business day during January 1983, the settlement position of all institutions before and after the removal of all payments to and from a large settling participant was calculated. For the associate participants of the settling participant, which were removed, it was assumed that they could find another settling participant or, if necessary, settle directly for themselves at the end of the day. The settling participant selected for the first simulation had a net credit position of $321 million for the day in which it was assumed to fail unexpectedly prior to settlement.

For the first revised settlement, 24 institutions had settlement obligations that had increased by more than the amount of their capital and ended up in a net debit position. Of these 24, 8 had been in a net credit position before the removal of the transactions involving the failed settling participant. Table 1 shows the original settlement position of each of these 24 institutions (column 1). From this and the revised settlement position (not shown), the magnitude of the settlement position change was computed, giving the ratio of the settlement position change to the institution's capital (column 2). In revising the settlement, the largest net position erosion was $568 million, which represented 83% of that institution's capital (and, since this is less than the institution's capital, this institution is not shown among the 24 in table 1). Even with this change, however, the institution remained in a substantial net credit position. The largest single net position enhancement was $762 million.

As already noted, there are a number of factors, including customer use made or permitted of nonsettled transfers, internal resources, the parent's comfort letter, borrowing, and coverage by a settling participant, that will influence a participant's ability to meet its revised settlement obligation. Certainly the interplay of these factors will vary from one participant to another. In order to pursue the settlement revision process to its ultimate conclusion over a number of iterations, the limitations of the study required a standard assumption. For simplicity, the same threshold as in table 1 was used: namely, if a participant experienced a position deterioration equal to or exceeding its capital and was in a net debit position in the revised settlement, then it too was presumed to be unable to settle. This series of iterations—one for each revised settlement—was continued until no participant failed by this definition in an iteration. Thus, for each iteration a revised net settlement position for all participants was computed after removing any participant that failed and all transfers to and from it. Table 2 shows the results of this process in the first simulation. It took 6 iterations before no participant failed to meet the above criteria. To the extent that institutions on average can sustain settlement position changes in excess of their capital, this table overstates the magnitude of the problem which could develop. If, on the other hand, an institution would have withdrawn from a settlement when its revised settlement obligation increases by more than 10 percent of its capital (rather than 100 percent), the impact is greater. For example, the 10 percent "rule" would make the last row in table 2 show 73 failed institutions (rather than 50) representing 76.1 percent of the total value of CHIPS messages sent that day (rather than the 38.6 percent).

One of the factors that has been mentioned that could facilitate achievement of a settlement is the willingness of a settling participant to represent one of its associated participants despite the latter's having difficulty meeting its revised settlement obligations. Table 3 groups the 24 institutions shown in table 1, illustrating the first iteration of table 2, by their settling participant, showing the aggregate net position change for the associates of each of the 10 settling participants who settle for other institutions. (The 12 settling participants who only settle for their own account are not shown.) The net position change for each of the settling participants is also shown. A contrast of columns 2 and 3 illustrates the impact on each settling participant of supporting all of its associate participants. In some cases the settling participant's net position is considerably worsened by folding in the associate's position with that of the settling participant.

A Second Simulation Using Transactions Data for a Different Day.

To assess the sensitivity of these simulation results to the transaction activity on the particular day used above, another day in January 1983 was randomly selected and the simulation analysis repeated. The results of this second simulation are summarized in tables 4-6. Overall, the results are broadly consistent with those for the first simulation examined. Somewhat fewer institutions (16 as opposed to 24) experienced an increase in their settlement position obligation that exceeded their capital and were in a net debit position for the revised settlement in the first iteration. In addition, the average ratio of position change to capital was somewhat lower for the second simulation. Two factors may explain these differences. First, the net position of the failed settling participant whose transaction activity was removed from the revised settlement in the second simulation was a very large credit balance. The removal of a settling participant in a large net credit position, compared to a small net credit or a net debit, was selected to approximate the situation where at least some adjustment by other participants in their exposures are made prior to the failure by the participant. For the second simulation, the net credit position of the settling participant removed was $921 million, roughly three times the same participant's net credit position for the first simulation ($312 million). A second reason could be that the total dollar value of messages sent on the day of the second simulation was significantly lower than on the day of the first simulation. The total value of messages sent in the first simulation was a fairly typical $255.6 billion. For the second simulation the total value of messages sent was only $89.4 billion or about a third of the earlier simulation.

One surprising fact that emerges in comparing the consequences of a failure of the same participant on two different days is the variability of other institutions affected. In view of the correspondent relationships among institutions, one might expect that those affected by a particular institution's failure would be fairly constant. In fact, there is a fair amount of variation. Of the 24 institutions in table 1 and the 16 in table 4, there were only 5 institutions in common in the first iteration. Similarly, of the total of 50 and 49 institutions in both tables 2 and 5, 16 of the institutions included in table 2 are not included in table 5. These results suggest that the institutions most likely to be affected by a particular institution's failure cannot be readily identified beforehand. This day-to-day variation in systemic risk exposure means that an institution cannot limit its credit assessment to some small group of participants with which it most frequently deals. Rather, it must carefully monitor all participants in each network and its exposure across multiple networks. It also means that from a supervisory standpoint the consequences of an institution's default on its payments obligations cannot be easily predicted.

A Third Simulation: The Settlement Failure of a Large Associated Participant.

In addition to analyzing the failure of a major settling participant, the consequences of the failure of a large associated participant on the settlement obligations of other participants was simulated. Tables 7-9 summarize the results obtained upon removing all transfers to and from a selected associated participant (the day studied is the same one considered in tables 1-3). While the number of participants experiencing an increase in their settlement obligation greater than their capital (and in a net debit position for the revised settlement) was somewhat less (13) than observed with the major settling participant (24) for the first iteration, the impact on the whole network is more similar than might be expected. Overall, 49 institutions failed to settle, accounting for 30.8 percent of that day's dollar value of messages sent. This similarity is noteworthy since the total dollar value of messages sent by this associated participant was only about 16 percent of the total value of the messages sent by the settling participant studied in the first simulation.

Repeating the simulation of the failure of a selected associated participant for the same day used in the second simulation resulted, after 8 iterations, in the failure of 33 institutions accounting for 22.6 percent of the total value of messages sent that day. The most significant aspect of this result is that the dollar value of all transfers sent by the associated participant for this second day represented only 0.8 percent of the CHIPS' total for the day.

Conclusions From the Simulation Exercise. The goal of the preceding simulations was to attempt to illustrate the possible implications for CHIPS participants of an unwinding of an institution's funds transfer activity due to one institution's sudden and unexpected default on its settlement obligation. As noted, there are a number of factors, such as the use institutions have made or permitted customers to make of nonsettled transfers, internal resources, the parents' comfort letter, borrowing, and coverage by a settling participant, that will decrease the likelihood of an unwind being necessary. But the simulations do raise serious, if not fundamental, questions about whether settlement revision using CHIPS Rule 13—the ultimate form of credit risk management for some wire transfer networks—is feasible without also disrupting financial markets. More importantly, the simulations illustrate the dynamic interdependence of institutions that characterizes the payment system. Overall, these simulations indicate that close to half of the participants and one-third of the payment value could be subject to the CHIPS unwind provisions in the event of a unexpected settlement failure of a CHIPS participant. This demonstrates that systemic risk can be significant and that its possible effects on financial markets may be large enough to be a serious problem.

It should be noted that these simulation results are conservative. This is because the "nonsettlement rule" used was (a) an institution experiences a reduction in its net credit or increase in its net debit that exceeds its capital and (b) the institution ends up on a net debit position in the revised settlement. Clearly, only (a) may be necessary for non-settlement. An institution may fail if it experiences a reduction in its net credit greater than its capital even if it still has a net credit in the revised settlement. In addition, an institution may decide not to settle its payment obligations that day if it would incur an apparent loss (at least before court action) of something much less than 100 percent of its capital. Thus, in these two important respects, the simulation exercise is clearly on the conservative side and, if anything, likely understates the impact of a settlement failure on the payments system and financial markets.

FOOTNOTES

5. This simulation analysis was performed by Jim Lyon (Federal Reserve Bank of Minneapolis) and Jack Walton (Federal Reserve Board), in addition to the author.

6. A failure to settle on CashWire has effects that actually seem to lie between CHIPS and FedWire, as is explained below when payments finality is discussed.

7. The customer, and perhaps a wider circle of the nonbanking economy, could be adversely affected, however.

Table 1

Failure of Major Settling Participant

Institutions with a Net Debit Position in the Revised Settlement and an Increase in Their Net Settlement Obligation Exceeding Their Capital

	Original Net Settlement Position (millions) (1)	Ratio of Position Change to Capital (2)
1.	$ 86.6	22.7
2.	18.5	21.4
3.	-16.8	1.5
4.	23.1	2.2
5.	-27.7	4.5
6.	-28.7	8.2
7.	-15.2	2.2
8.	-71.8	1.7
9.	-15.8	19.1
10.	-160.7	1.8
11.	3.5	4.2
12.	-4.4	8.4
13.	-21.5	15.8
14.	40.8	32.4
15.	-25.8	7.5
16.	0	12.4
17.	-9.1	7.1
18.	-2.7	3.0
19.	2.0	11.1
20.	-37.8	1.4
21.	-378.5	2.9
22.	6.6	7.7
23.	-39.0	12.1
24.	177.3	15.4

Table 2

Failure of Major Settling Participant

Iteration	Institutions with a Net Debit Position in the Revised Settlement and an Increase in Their Net Settlement Obligation Exceeding Their Capital (1)	Percent of the Daily Total Dollar Value of Messages Sent (2)
1	24	15.2
2	12	14.4
3	10	6.2
4	3	2.7
5	1	0.1
6	0	—
Total	50	38.6

Table 3

Failure of a Major Settling Participant

Revised Settlement Position After the Failure of a Participant for Institutions that Settle for Other Participants and the Increase in Settlement Obligation Due to Associated Participants

	Settling Participant's Original Net Position (millions) (1)	Settling Participant's Revised Net Position (millions) (2)	Aggregate Settlement Obligation Increase of Associated Participants (millions) (3)
1.	$ 52.3	$ 13.4	$ -7.4
2.	545.9	284.1	—
3.	147.2	99.6	-95.5
4.	-846.9	-138.8	-57.6
5.	106.9	868.7	-690.6
6.	1,132.8	564.7	-102.3
7.	44.2	347.3	-159.3
8.	22.6	-229.7	—
9.	-549.6	-608.5	—
10.	*	—	-9.6

---

*Failed settling participant.

Table 4

Failure of Major Settling Participant (Day 2)

Institutions with a Net Debit Position in the Revised Settlement and an Increase in Their Net Settlement Obligation Exceeding Their Capital

	Original Net Settlement Position (millions) (1)	Ratio of Position Change to Capital (2)
1.	$ 9.1	4.7
2.	-60.3	13.4
3.	16.6	5.8
4.	5.3	1.2
5.	-23.0	3.0
6.	13.9	3.5
7.	-16.6	1.8
8.	58.1	4.3
9.	-11.0	1.3
10.	0.2	1.5
11.	-75.4	4.7
12.	-16.6	1.9
13.	2.0	2.8
14.	-56.0	3.6
15.	-91.0	1.2
16.	-2.6	103.4

Table 5

Failure of Major Settling Participant (Day 2)

Iteration	Institutions with a Net Debit Position in the Revised Settlement and an Increase in Their Net Settlement Obligation Exceeding Their Capital (1)	Percent of the Daily Total Dollar Value of Messages Sent (2)
1	16	8.7
2	11	9.4
3	12	11.3
4	9	2.1
5	1	0.1
6	0	—
Total	49	31.6

Table 6

Failure of a Major Settling Participant (Day 2)

Revised Settlement Position After the Failure of a Participant for Institutions that Settle for Other Participants and the Increase in Settlement Obligation Due to Associated Participants

	Settling Participant's Original Net Position (millions) (1)	Settling Participant's Revised Net Position (millions) (2)	Aggregate Settlement Obligation Increase of Associated Participants (millions) (3)
1.	$ 148.9	$ 135.8	—
2.	527.6	453.6	$ -40.9
3.	288.0	266.2	-13.4
4.	17.3	1,026.0	-68.8
5.	-1,076.0	-686.4	-151.7
6.	568.9	735.4	-16.8
7.	-148.9	-544.8	-2.7
8.	-140.4	-184.0	—
9.	-152.2	-107.7	—
10.	*	—	-75.0

---

*Failed settling participant.

Table 7

Failure of Large Associated Participant

Institutions with a Net Debit Position in the Revised Settlement and an Increase in Their Net Settlement Obligation Exceeding Their Capital

	Original Net Settlement Position (millions) (1)	Ratio of Position Change to Capital (2)
1.	$ -40.5	*
2.	84.5	225.1
3.	-16.8	2.2
4.	23.1	1.7
5.	-28.9	1.4
6.	-380.9	1.6
7.	4.2	2.1
8.	-257.2	1.4
9.	-15.8	3.5
10.	21.9	2.3
11.	-231.4	1.1
12.	-9.1	15.9
13.	8.7	46.0

---

*Institution for which capital figure not available from Board of Governors' data.

Table 8

Failure of Large Associated Participant

Iteration	Institutions with a Net Debit Position in the Revised Settlement and an Increase in Their Net Settlement Obligation Exceeding Their Capital (1)	Percent of the Daily Total Dollar Value of Messages Sent (2)
1	13	11.6
2	18	8.4
3	10	4.9
4	4	3.7
5	2	1.4
6	1	0.4
7	1	0.4
8	0	--
Total	49	30.8

Table 9

Failure of a Large Associated Participant

Revised Settlement Position After the Failure of a Participant
for Institutions that Settle for Other Participants and the
Increase in Settlement Obligation Due to Associated Participants

	Settling Participant's Original Net Position (millions) (1)	Settling Participant's Revised Net Position (millions) (2)	Aggregate Settlement Obligation Increase of Associated Participants (millions) (3)
1.	$ 52.3	$ 42.4	—
2.	311.9	617.6	$ -19.3
3.	545.9	640.0	-18.4
4.	147.2	-4.3	-225.1
5.	-846.9	-573.2	-25.7
6.	106.9	141.6	—
7.	1,132.8	1,130.1	-34.4
8.	44.2	-90.6	-43.1
9.	22.6	-29.6	-28.5
10.	-549.6	-580.1	—

Chairman **Fauntroy**. Very good. Thank you very much. We appreciate the candor with which you have addressed these subjects, which are of concern to us, and one which we are going to have to continue to work very hard in a number of areas.

Mr. **Corrigan**. Thank you, Mr. Chairman.

Chairman **Fauntroy**. Our final witness today is from the financial institution which had this problem. I would like to welcome Mr. J. Carter Bacot, chairman and chief executive officer of The Bank of New York.

Mr. Bacot, thank you for coming.

STATEMENT OF J. CARTER BACOT, CHAIRMAN AND CHIEF EXECUTIVE OFFICER, THE BANK OF NEW YORK, ACCOMPANIED BY DENO PAPAGEORGE, SENIOR EXECUTIVE VICE PRESIDENT, CHIEF FINANCIAL OFFICER, AND SHELDON HARRISON, SENIOR EXECUTIVE VICE PRESIDENT, CHIEF OPERATIONS OFFICER

Mr. **Bacot**. Thank you, Mr. Chairman. I am pleased to be here to discuss the problem that arose.

First, let me introduce my two colleagues: Mr. Deno Papageorge on my right is senior executive vice president and chief financial officer of the bank. The government securities department is in his area of responsibility. On my left is Sheldon Harrison, senior executive vice president and chief operations officer of the bank.

Let me start by saying The Bank of New York is the Nation's second oldest bank, and currently is the 18th largest bank in the United States. We are in our 201st year of operation providing a full range of banking services to businesses and consumers. Our parent company has a primary capital-to-assets ratio of 7.71 percent, which is one of the strongest among large bank holding companies in the country.

Before turning to a detailed explanation of what happened on November 21, let me summarize what did not happen. There was no credit risk for The Bank of New York or the Federal Reserve, there was no computer capacity problem, and no loss of computerized data, there was no financial loss to taxpayers, investors, or consumers; there was no financial loss to Government securities dealers or the Federal Reserve, and there was no effect on the ability of the Treasury to finance the Government. What happened, in one sentence, was that a flaw in the computer program design—I don't know if "flaw" is better than "glitch," but we term it a "flaw"—caused a delay in book-entry delivery of Government securities for which we paid a penalty of $5 million. We will not recover this cost.

The Bank of New York's Government securities clearance service, or GSCS as it is commonly known in the trade, is a highly sophisticated, computerized system by means of which The Bank of New York has, since 1977, acted as a clearing agent for the settlement of purchases and sales of the full range of Government securities.

It is important to distinguish between dealers in Government securities, who buy and sell such securities, and clearing agents, who process transactions for dealers. It should be emphasized that, in its role as a clearing agent for Government securities, The Bank of New York assumes no market risk.

Currently, the customers of GSCS include 40 broker-dealers. These customers are carefully selected by The Bank of New York on the basis of their credit worthiness, the strength of their operations and their reputation in the financial community. Many applicants for participation in GSCS have been turned down because they failed to meet The Bank of New York's rigorous criteria. Among the applicants to GSCS whose applications were turned down were E.S.M. Government Securities, Inc., Bevil Bressler & Schulman Asset Management Corp., the Lion Capital Group, Drysdale Government Securities and Comark Securities. We turned them down because they failed to meet our standards and criteria.

Every day we clear thousands of Government security transactions. The clearing function performed by the bank and other major clearers—as has been mentioned before—is essential to the maintenance of an orderly market in the Government securities business. It provides valuable assistance to the Treasury and the Federal Reserve in facilitating the placement of the Government debt. Highly complex computers are required to perform the function. It involves substantial outlays of capital and necessitates the hiring of trained and sophisticated computer personnel.

The Bank of New York is one of the two largest clearers of Government securities today. We believe our success is due to our superior computer system and our superior service. It is noteworthy that the computer problem we are going to discuss here has had no discernible effect on our customer relationships. In fact, since the incident of November 21, one of the largest continental European banks and one major U.S. Government securities dealer have decided to move their business to the bank.

As I mentioned, the computer problem on the 21st and the delay were caused by a flaw in the program design. The Bank's Government Securities Program has been in place for several years and worked well. It was revised in March to permit the computer to recognize and store information for approximately 36,000 Government securities issues, well in excess of the number of Government securities issues in the market as of November 20 and 21. It is our continuing goal to anticipate the ever increasing volume in this business. Due to a flaw in the program design, however, there were inadequate instructions regarding the method of storing data when the number of issues cleared exceeded 32,000. On November 20, the number of issues cleared did exceed that 32,000 level for the first time. Early on the 21st, lacking adequate storage instructions for additional data, the computer stored the data incorrectly and corrupted the data base. The close of trading on Wednesday, November 20 was quite late. It was about 10 o'clock because of a computer problem at another institution. This delayed our preparation of reports and statements on Wednesday's work and thus delayed our recognition that a problem existed with respect to the Wednesday's data.

At 10 a.m. on Thursday, our employees performed their customary manual check of customer reports and saw something was wrong with the data. They immediately notified the Federal Reserve. Two of our computer programmers and three supervisory officials, as well as a large support staff, began working full-time to identify the source of the problem. At about 12:30 or 1 o'clock, they identified it as the flaw in the program I have described. At this point the bank's special outside computer consultants were called in.

The next task was to figure out how to correct the problem as quickly as possible with a minimum of disruption to the ongoing clearing system. The approach we decided to take was the standard approach taken in the computer industry for a software problem. It is one that has been used in every instance of computer failure in the past in securities clearing and one that has always worked.

First, we assigned a group of experts to the task of coming up with a temporary solution. This is known in the industry as a patch, similar to a patch on a tire to keep it going until you can deal with the problem more thoroughly later on. This would enable the system, as the tire analogy suggests, to continue to function.

Second, and at the same time we assigned a separate group of experts to the task of developing and implementing an entire program redesign. That is the longer term solution. This redesigned program would clean up the corrupted data from Wednesday and replace the old program that had caused the problem.

We knew from the outset that redesigning the program would mean dealing with a review of 700 separate programs within the system and would take approximately 16 hours. Therefore, it could not be completed before the Federal Reserve's securities wire had to close to go over to the next day's processing. We could not have completed the long-term solution within that period of time.

We had every reason to believe that the short-term patch solution would bridge the gap and solve the problem. So throughout Thursday, we pursued both projects with maximum effort. Members of senior management of the bank were informed and were in direct contact with the Federal Reserve bank into the early morning hours on Friday. Ultimately some 55 people were involved in working on the problem.

Throughout the day Thursday patches were developed and tested. At 8 p.m. on Thursday we developed a patch that was able to process trades, but at a slower than normal pace. It gave out at about 12:45 a.m. Friday morning. Shortly thereafter, the Federal Reserve, as you know, closed the wires. During the 4½ hours that the patch operated, we were able to clear a little over $8 billion of securities. This reduced the problem from the high of approximately $31 billion, which has been mentioned before, to the $22.6 billion. Thus we submit it was in the interest of all concerned to keep the wire open and run the patch computer program for as long as possible to mitigate the problem. The patch effort in no way delayed or compromised the long-term effort to develop and implement a full new system redesign. Moreover, the decision not to halt clearance of Government securities despite the delays resulted in far less disruption to the market than would otherwise have occurred.

Although the Federal securities wire closed at 1:30, we continued our full-scale efforts throughout the rest of the night. As anticipated, we completed our Wednesday and Thursday processing work by 10 a.m. Friday morning. By shortly after noon on Friday, we began to process that day's normal trade. That was a function of the longer term solution, which took 12 hours instead of 16 hours, working on the program redesign. Our computer started processing, and did process on Friday, 1¾ day's worth of business. Our computer had no further problems and the Federal Reserve loan was repaid that day.

At no point in this course of events was The Bank of New York in financial jeopardy. The situation that led to the necessity of the overnight borrowing from the Federal Reserve discount window differs fundamentally from other well publicized bank borrowings. It differs in that this was fully collateralized by the bank's interest in highly liquid and highly marketable government securities that it could not deliver and was repaid in full with interest on the following day.

The problem really was one of timing and not of credit, not in any way of credit. Because of the computer program problem we had Government securities in effect rather than cash. We had the assets of the same value as if everything had functioned normally, but because these assets were in the form of Government securities and not cash, we could not balance our books at the Federal Reserve.

As a result, when the securities wire closed, we requested the overnight loan from the Fed to cover our obligation. We paid $5 million for this risk-free loan fully collateralized by Government securities of equal value.

Also, I should note here our customers received the full amount of interest that had accrued on their securities during the time that the bank was unable to deliver them. In no way did they lose any money. The Bank of New York, by law, cannot collect interest on those securities while it has them in its possession.

I cannot see the basis for suggesting that, beyond our payment of $5 million in interest to the Federal Reserve, some additional punitive measure should be taken against the bank. Indeed, $5 million represents a very substantial penalty for our stockholders, for our officers, and for all of our employees.

We spend literally hundreds of hours every year working to improve and create greater safeguards for the system. Our Government security clearance system has currently a complete backup computer system and a staff of five full-time programmers dedicated just to this system. Our system has been so well thought of in the past that it has been used as a model by some Federal Reserve banks.

Nonetheless, we are deeply concerned about the recent problem. We have taken immediate steps to insure that there is no repetition of it. This includes, first, the formation of an internal task force composed of senior bank officers to study the incident; second, extensive consultation with the Federal Reserve bank and the other clearing banks—this has already started—third, a review of both equipment and personnel in our Government securities clearance operations; fourth, the installation of an improved diagnostic system; and, fifth, the employment of special outside consultants.

I want to emphasize in closing what we did have on November 21 was a limited operational failure, which was resolved within 24 hours, with no financial loss to anyone except The Bank of New

York. Thus it failed briefly, but it failed safely and safely in the sense there was no disruption of the Government securities market and no loss to anyone, as I mentioned, but ourselves.

Nevertheless, we should not, and no one else should, be complacent about this kind of situation or complacent about the future. We will continue our efforts with the industry and with the Federal Reserve and with others to improve the safety and the soundness of the System. That concludes my remarks, Mr. Chairman.

[The prepared statement of Mr. Bacot on behalf of The Bank of New York; a document entitled "Consolidated Reports of Condition and Income for A Bank With Domestic and Foreign Offices—FFIEC 031"; an excerpt from Moody's Bank and Finance Manual; and the 1984 Annual Report of The Bank of New York, follow:]

STATEMENT OF J. CARTER BACOT, CHAIRMAN AND CHIEF EXECUTIVE OFFICER OF THE BANK OF NEW YORK AND THE BANK OF NEW YORK COMPANY, INC.

BEFORE
THE SUBCOMMITTEE ON DOMESTIC MONETARY POLICY
OF THE COMMITTEE ON BANKING, FINANCE AND
URBAN AFFAIRS

UNITED STATES HOUSE OF REPRESENTATIVES
December 12, 1985

Mr. Chairman, honorable members of the Subcommittee, ladies and gentlemen. My name is J. Carter Bacot. I am the Chairman and Chief Executive Officer of The Bank of New York and its parent, The Bank of New York Company, Inc.

I appreciate this opportunity to appear before the Subcommittee to respond to the Subcommittee's questions concerning the recent computer programming problem at The Bank of New York's government securities clearance operation and to explain what happened, why, what we did and what we are doing about it. During the course of this statement I hope to answer each of the specific questions raised in the Chairman's letter.

The Bank of New York is the nation's second oldest and 18th largest bank. It is now in its 201st year of operation. Its parent company has a primary capital to assets ratio of 7.71%, which is one of the strongest among large bank holding companies in the country. We provide a full range of banking services to business and consumers. On the consumer side, for example, our banking organization is one of the largest credit card and automobile lenders in New

York because it offers the lowest interest rates of any major commercial banking organization in New York City.

The Bank of New York is one of the leading processors of securities transactions in the United States. Its business includes servicing mutual funds, unit investment trusts and dividend reinvestment plans, acting as custodian of securities for institutions and individuals, acting as dividend paying agent, trustee, registrar and transfer agent for corporations, and clearing government securities, which is the business to be discussed today.

Before turning to a detailed explanation of what happened on November 21, let me summarize what did not happen: there was no credit risk for The Bank of New York or for the Federal Reserve; there was no computer capacity problem and no loss of computerized data; there was no financial loss to taxpayers, investors or consumers; there was no financial loss to government securities dealers or the Federal Reserve; and there was no effect on the ability of the Treasury to finance the Government. What happened, in one sentence, was that a flaw in the design of a computer program caused a delay in book-entry delivery of government securities, for which The Bank of New York paid a penalty of $5 million.

GOVERNMENT SECURITIES CLEARANCE SERVICE.

The Bank of New York's Government Securities Clearance Service, or GSCS as it is commonly known in the trade, is a highly sophisticated computerized system by means of which The Bank of New York has, since 1977, acted as a clearing agent for the settlement of purchases and sales of the full range of government securities. In short, a clearing agent services buyers and sellers of securities by processing the transfer of securities from buyers to sellers. At one time this function was performed manually, but because of the rapid growth of the market and the Federal Reserve's development of the computer based book-entry system, clearance is now performed almost entirely by computer.

It is important to distinguish between dealers in government securities, who buy and sell such securities, and clearing agents, who process transactions for dealers. It should be emphasized that, in its role as a clearing agent for government securities, The Bank of New York assumes no market risk.

Currently, the customers of GSCS include forty broker-dealers. These customers are carefully selected by The Bank of New York on the basis of their creditworthiness, the strength of their operations and their reputation in the financial community. Many applicants for participation in GSCS have been turned down because they failed to meet The

Bank of New York's rigorous criteria. Among the applicants to GSCS whose applications were turned down were E.S.M. Government Securities, Inc., Bevil Bressler & Schulman Asset Management Corporation, The Lion Capital Group, Drysdale Government Securities and Comark Securities.

The government securities market works largely on a book-entry basis. In the book-entry system there are no printed certificates evidencing ownership of a government security, and hence there is no physical delivery of securities. Instead, ownership and transfer of ownership of government securities are recorded on computers. This book-entry system was developed by the Federal Reserve and the Treasury Department.

Let me provide an illustration of how a typical clearance transaction occurs. Assume that Dealer A sells U.S. Treasury Notes to Dealer B. On the settlement date, Dealer A instructs its clearing bank, Bank X, to deliver the securities to The Bank of New York, which is the clearing bank for Dealer B.

The following steps now occur: (1) Bank X instructs the Federal Reserve Bank of New York to transfer the Treasury Notes from Bank X's securities account at the Reserve Bank to The Bank of New York's securities account at the Reserve Bank, (2) the Reserve Bank credits Bank X's deposit account for the value of the securities, (3) the Reserve Bank charges The Bank of New York's deposit account

in the same amount, (4) The Bank of New York "delivers" the securities according to the instructions of Dealer B, (5) Dealer B pays The Bank of New York, and (6) The Bank of New York repays any outstanding obligation to the Federal Reserve. By means of the GSCS computer system the settlement of such a trade is, under normal circumstances, virtually instantaneous.

Every day thousands of transactions like this one clear GSCS. The clearing function performed by The Bank of New York and other major clearers is essential to the maintenance of an orderly market in government securities. It provides valuable assistance to the Treasury and the Federal Reserve in facilitating the placement of government debt and significantly reduces the pressure on the Federal Reserve by matching internally up to 35% of the purchases and sales, processing that would otherwise have to be done by the Federal Reserve itself. Because of the high volume of transactions, the system simply cannot be run manually. Not only are computers necessary, but highly complex computers are required, involving substantial capital outlays and necessitating the hiring of highly trained computer personnel.

The Bank of New York is one of the two largest clearers of government securities in the country. We believe this success is due to our superior computer system and superior service. For example, we recently set a record in clearing 29,000 government securities transactions in one day. We provide our customers with the safest and most efficient system available and have pioneered the use of numerous cost and time saving innovations. Indeed our reputation among government securities brokers and dealers is such that the computer failure that we are here to discuss today has had no discernible effect on our customer relationships. In fact, since the incident on November 21, one of the largest continental European banks and one major U.S. government securities dealer have decided to move their business to the Bank. Moreover, many of our customers have called to express concern that this event might cause us to scale back our activities as a clearing agent. They have encouraged us to continue in our role as a leading government securities clearing facility.

THE COMPUTER PROBLEM ON NOVEMBER 21.

The computer problem on November 21, and the delay in processing transactions that it occasioned, was caused by a flaw in a computer program design, and was exacerbated by another bank's computer problem on November 20. I will provide an overview of the events on those dates. A complete chronology is attached as Exhibit A.

The Bank's GSCS computer program had been in place for several years and had worked well. It had been revised in March to permit the computer to recognize and store information about approximately 36,000 government securities issues, well in excess of the number of government securities issues in the market as of November 20 and 21. Due to a flaw in the program design, however, there were inadequate instructions regarding the method of storing data when the number of issues to be cleared exceeded 32,000. On November 20, the number of issues being cleared exceeded 32,000 for the first time. Lacking adequate storage instructions for the additional data, the computer stored the data incorrectly and corrupted the data base.

After the close of clearing every day, the GSCS computer generates numerous reports and statements for customers. The close of clearing on November 20, 1985 was very late, about 10:00 p.m., because of a computer problem at another institution. This delayed the preparation of reports and statements of Wednesday's work and, ultimately, the recognition of the problem.

At 10:00 a.m. on Thursday, November 21, employees at The Bank of New York performed their customary manual check of the reports generated by the computer for customers summarizing Wednesday's trades, and saw that something was wrong. They notified the Federal Reserve. Two GSCS computer programmers and three supervisory officials, as well as a support staff, began working full time to identify the source of the problem. By about 12:30 p.m. they identified the problem as the flaw in the program design I have described. At this point, the Bank's special outside computer consultants were called in.

The next task was to figure out how to correct the problem as quickly as possible with a minimum of disruption to the ongoing clearing system. The approach we decided to take is standard in the computer industry for software problems. We assigned one group of experts the task of trying to come up with a temporary solution, known in the industry as a "patch", like a patch on a tire. This patch would enable the clearing system to continue to operate and in the past had always worked. At the same time, we assigned a separate group of experts the task of developing and implementing an entire program redesign. This redesigned program would clean up the corrupted data from Wednesday and replace the old program that had caused the problem. We knew from the outset that redesigning the program would require the review of 700 separate programs, would take approximately 16 hours and therefore could not be completed by the time the securities wire closed at the Federal Reserve. We had every reason to believe, however, that a patch could be developed in the meantime to clear the trades by the close of the wire.

All day Thursday both projects continued with maximum effort. Members of senior management of the Bank were informed and were in direct contact with the Federal

Reserve Bank. Ultimately, some 55 people at the Bank were involved in working on the problem.

Throughout Thursday, patches were developed and tested. At 8:00 p.m., we developed a patch that was able to process trades, but at a slower than normal pace. It gave out at about 12:30 a.m. Friday morning. Shortly thereafter the Federal Reserve closed its securities wire. During the 4 1/2 hours the patch operated, however, the Bank was able to clear over $8 billion of securities. This reduced the strain on the overall system and reduced the Bank's debt to the Federal Reserve by a similar amount.

The patch effort in no way delayed or compromised the effort to develop and implement the full scale program redesign. Moreover, the decision not to halt clearing in government securities, despite the delays occasioned by the computer problem at The Bank of New York, resulted in far less disruption to the market than would otherwise have occurred.

Although the Fed securities wire closed at 1:00 a.m., the Bank continued its full scale efforts during the remainder of the night. As anticipated, the Wednesday and Thursday processing work was completed by 10:00 a.m. Friday morning. By shortly after noon on Friday, we began processing that day's trades on a normal basis. On Friday we processed one and three-quarters day's worth of business.

Our computer had no further problems. The Federal Reserve loan was repaid that day.

THE PROBLEM WAS NOT A CREDIT PROBLEM; IT WAS A SETTLEMENT PROBLEM.

At no point in this course of events was The Bank of New York in any financial jeopardy. The situation that led to the necessity of an overnight borrowing of $22.6 billion from the Federal Reserve discount window, fully collateralized by the Bank's interest in the government securities that it could not deliver, differs fundamentally from other well publicized bank borrowings from the Federal Reserve. The problem here was one of timing, not credit.

Let me refer to my example of a transaction involving a purchase of government securities by Dealer B from Dealer A. When our computer problem arose, the Bank was able to, and did, receive securities from Dealer A through A's clearing agent, Bank X, and the Federal Reserve Bank paid Bank X and charged our deposit account. As a result of the computer problem, however, The Bank of New York was not able to deliver the securities to our customer, Dealer B, until Friday and therefore could not receive payment from Dealer B until Friday. In the meantime, after thousands of these partially cleared transactions, The Bank of New York held almost $25 billion in government securities at the end of the day on Thursday, and had a $22.6 billion obligation to the Reserve Bank.

If the computer system had been functioning normally, the Bank would have delivered out these securities, received cash payments, and made corresponding payments to the Reserve Bank. Because of the computer problem, however, the Bank had government securities rather than cash, and the system as a whole had cash rather than government securities. For every security held by the Bank, there was a matching delivery instruction that would generate the cash to repay the Reserve Bank. The Bank had assets of the same value as if everything had functioned normally, but because these assets were in the form of securities and not cash, it was unable to balance its deposit account at the Reserve Bank, which is in cash. This resulted in the formalization of the Bank's obligation into a $22.6 billion overnight loan from the Federal Reserve at the normal discount rate of 7.5%. Again, I would stress that this loan was fully collateralized by the Bank's interest in government securities of greater value.

This subcommittee has inquired why the Bank borrowed through the discount window rather than engaging in a repurchase agreement with the Reserve Bank. The discount window is the normal method of bank borrowing from Reserve Banks under Sections 201.3(a) and 201.4(a) and (b) of Federal Reserve Board Regulation A.

This subcommittee has also inquired why The Bank of New York did not transfer its position to another clearing bank and did not borrow from the private sector rather than the Reserve Bank. The same computer problem that prevented delivery of securities on Thursday precluded any transfer to another bank. The answer to the borrowing question again relates to timing. The securities wire closed at 1:00 a.m. in the morning. At that time, there was very limited availability of interbank funds, although the Bank did make every effort to borrow such funds and did borrow $700 million.

Due to the fact that The Bank of New York could not rectify the results of an unforeseen and probably unforeseeable flaw in a computer program prior to the closing of the securities wire at 1:00 a.m., The Bank of New York sustained a loss of approximately $5 million, the interest on the discount window borrowing. We were the only real financial loser in the entire incident. Customers received the full amount of the interest that accrued on their securities during the time the Bank was unable to deliver them.

THE PROBLEM WAS OPERATIONAL AND IS NOT SYMPTOMATIC OF ANY FUNDAMENTAL PROBLEM IN THE CLEARING SYSTEM.

As I hope my explanation has made clear, the computer problem at The Bank of New York was the result of a flaw in program design that led to a significant, and, for the Bank, costly, delay in the settlement of government securities transactions. The problem was not in any sense a disaster or even a near disaster, nor is it symptomatic of any fundamental problem with the clearing system.

Some press reports have suggested that there was a system overload, but this is not the case. It was not the volume of trades that led to the computer failure. The Bank of New York's computer is capable of handling, and has handled, a volume several times greater than the volume it handled on November 20. The problem arose out of a flaw in program design, which has now been corrected.

No computer system is perfect. It is a fact of life in this computer age that thousands of computers go down every day for countless reasons. Computers go down not only at the bank clearers, but at securities dealers and the Federal Reserve banks. If the government securities clearing system were to halt every time one of its member's computers went down, the system could not function. Instead, participants cooperate when computer problems occur to minimize disruptions to the system. In every instance prior to November 21, the problem had been rectified on a timely basis.

Ironically, in this case, The Bank of New York's computer system, considered by many to be the best computer system of its kind, was not able to complete delivery of securities by the close of the securities wire. As a consequence, under the current scheme, we were forced to borrow money to cover this overnight, computer generated debt and to pay what amounts to a $5 million penalty for it. Under these circumstances, I cannot see the basis for suggesting that, beyond this payment of interest to the Federal Reserve, some additional punitive measure should be taken against The Bank of New York. Indeed, $5 million represents a very substantial penalty for our stockholders, officers and employees.

The participants in the government securities clearing system work constantly to improve and create greater safeguards for the system. We spend literally hundreds of hours every year meeting to discuss improvements, and a larger number of hours to implement them. The Bank of New York currently has a complete back-up computer system and a staff of full-time computer programmers. Our system is so well thought of that it is used as a model by some Federal Reserve banks.

We are deeply concerned with the recent computer problem. We have taken immediate steps to attempt to ensure that there will be no repetition. This includes the formation of an internal task force, composed of senior Bank officers, to study the incident and make recommendations; intensive consultation with the Federal Reserve Bank of New York and the other clearing banks; review of both equipment and personnel in our GSCS operations; installation of an improved diagnostic system; and the employment of special outside consultants.

I want to emphasize that what we had was a limited operational failure, which was resolved within 24 hours, without loss to any one except the Bank. The resolution of this problem illustrates, to my mind, not the existence of fundamental problems in the government securities clearing system, but the fundamental soundness of the system.

POSSIBLE CHANGES IN THE OPERATION OF THE GOVERNMENT SECURITIES MARKET.

This subcommittee has asked several questions regarding possible changes in the operation of the government securities market. First, regarding the frequent delays in closing, we think that the problem is not the scheduled closing hour, but rather the need to distribute the clearing traffic more evenly throughout the day. The Federal Reserve, the dealers and clearing banks should work together to develop an effective remedy in this regard.

Second, we recommend that the free enterprise system should continue to prevail in the clearing business. Once again, it is important to recognize that, despite the problem on November 21, the Bank was in no danger, the Federal Reserve and the Treasury were in no danger and the dealers, the investors and the public at large were protected. We question whether an increase in the number of clearing banks would improve operations of the market, because such a development is likely to result in increased potential for computer errors. The existence of only a relatively few significant clearing banks reflects market practice and conditions, in particular the high cost of developing complex, sophisticated computer systems and training skilled groups of computer experts and credit analysts. At the same time, however, any bank that wishes to devote sufficient resources to becoming a clearing bank can do so. The present clearing system is highly efficient and capable of handling the great number of transactions which occur today to enable our Government to finance itself.

Exhibit A

CHRONOLOGY RELATING TO COMPUTER PROBLEM AT THE BANK OF NEW YORK

Wednesday November 20	
2:30 p.m.	Scheduled closing time for the Federal Reserve Book Entry securities wire.
10:00 p.m.	Actual close of Securities Wire due to computer problems of another major clearing bank; usual end of day balancing of customer cash positions begins.
Thursday November 21	
1:00 a.m.	Completion by The Bank of New York ("BNY") of customer cash proving procedures for work of November 20. (Normal completion time is three hours after wire close.)
8:00 a.m.	Frederick Roemer, Vice President and head of the Government Securities Clearance department, spoke with Daniel Bolwell of Federal Reserve Bank of New York ("FRB") and advised him that BNY's Wednesday nightly update of customer data files and reports still being run.
10:00 a.m.	Day begins late due to late close of securities wire. Completion of customer position reports. When the Bank conducts customary manual examination procedures, Frederick Roemer detects a problem of corrupted data for Wednesday's trades.
10:00 a.m. to 11:30 a.m.	Programmers are informed by Frederick Roemer of problem and commence attempts to find source of problem. Robert Sontag, Assistant Vice President and assistant head of the Government Securities Clearance department of BNY and Daniel Bolwell speak on hotline at 10:15 a.m. about the existence of a problem. From that time on BNY and FRB in contact at least as frequently as every half hour. Diagnostic procedures begin. Two programmers, three managers and a support staff at work full time on problem.
11:30 a.m. to 1:00 p.m.	Steven Jukofsky, Vice President and head of the Securities Clearance Group, is informed of problem and begins participating in diagnosis and development of solutions. Problem diagnosed as flaw in program. Informed FRB of seriousness of problem. Additional computer experts from outside software development firm, which developed the program, are called in.

1:00 p.m. to 2:00 p.m.	Steven Jukofsky and Frederick Roemer make decision, in keeping with normal operating procedures in the event of programming problems, to pursue two approaches simultaneously: (1) to develop a temporary "patch" that will enable the Bank to clear transactions for the day before close of the securities wire Thursday, and (2) to develop a full scale program revision, projected to require approximately 16 hours of work, and therefore not able to be completed by the close of the securities wire. For the rest of Thursday and through the early morning hours of Friday, two teams of programmers work around the clock to develop structures for both these solutions.
2:00 p.m. to 2:30 p.m.	Rosanne Witmer, Senior Vice President and head of Securities Industry Division, is informed of problem. She reviews and concurs in strategy, attempts to estimate time required to develop a patch and speaks with Carol Barrett of the FRB to inform her of status of efforts to patch program. First patch attempt fails. From then on, at approximately hourly intervals, patches are developed and tested.
2:30 p.m. to 4:00 p.m.	Rosanne Witmer speaks with Jorge Brathwaite, Vice President, Operations, at FRB and gives him an update. She asks for an extension to complete the patch. From then on, throughout the incident, the two are in constant telephone contact. Rosanne Witmer leaves message to inform Deno Papageorge, Senior Executive Vice President and Chief Financial Officer of BNY, of problem.
4:00 p.m. to 5:00 p.m.	Rosanne Witmer informs J. Carter Bacot, Chairman and Chief Executive Officer of BNY, and Peter Herrick, President of BNY, of problem. From that point forward, until 1:30 a.m. Friday, the President is kept constantly informed and participates actively in decision-making. Deno Papageorge returns from meeting, is briefed on status of effort to solve problem, and speaks with Thomas Timlen, First Vice President at FRB, and informs him that patch will probably be operating between 5:30 and 6:30. Additional senior programmer, who is president of the software development firm and has been constantly in contact by phone, arrives on site.

5:00 p.m. to 6:00 p.m.	Rosanne Witmer tells Jorge Brathwaite that patching has proved more difficult than initially expected and projects that system will be up at 7:00 p.m. Brathwaite suggests a 2 hour extension. Deno Papageorge tells Thomas Timlen that BNY projects that programming work on patch will be completed by 7:00 p.m. Rosanne Witmer and three other commercial banking officers responsible for securities industry business, Michael Hurley, Vice President, Giovanni Fallone, Assistant Vice President, and William Tomai, Assistant Treasurer, come on site to monitor risk and coordinate communications with FRB, senior management and customers.
6:00 p.m. to 7:00 p.m.	Rosanne Witmer discusses security agreement in event of loan from FRB with Susan Sibel in BNY legal department and with legal department of FRB. Legal department at FRB sends draft of security agreement to legal department at BNY. Deno Papageorge calls Thomas Timlen to tell him that additional system adjustments will delay opening until 8:00 p.m.
8:00 p.m.	GSCS computer comes up. BNY liability to FRB $32 billion. FRB agrees to continuing extensions. Deno Papageorge signs FRB loan documents.
10:00 p.m.	Carter Bacot and E. Gerald Corrigan, President of FRB, begin telephone communications which continue off and on until 1:30 a.m.
12:00 midnight to 12:30 a.m.	Gerald Corrigan informs Carter Bacot that FRB cannot grant time extension beyond 1:00 a.m.
Friday November 22	
12:30 a.m.	Patch gives out and computer goes down after clearing $8 billion in securities; BNY purchases $700 million in Fed Funds from the private sector.

1:00 a.m.	FRB closes the securities wire.
1:30 a.m.	FRB closes the securities wire for reversals.
2:00 a.m.	Development of full scale program redesign completed. Programmers submit schedule to management to reconstruct and correct Wednesday and Thursday's customer data and be ready for Friday clearing by noon Friday. Management approves schedule and plan. Rosanne Witmer, Steven Jukofsky, Frederick Roemer, Robert Sontag, Michael Hurley, Giovanni Fallone, and William Tomai remain on site to monitor progress throughout the night.
2:15 a.m.	FRB closes money wire. BNY requests discount window loan.
5:00 a.m.	Reconstruction and correction of Wednesday customer data files completed.
7:00 a.m.	Additional BNY operation staff arrive at Bank.
8:00 a.m.	Carter Bacot, Peter Herrick, Deno Papageorge, Rosanne Witmer, James Matthews (Vice President and Counsel of BNY), Frederick Roemer and Steven Jukofsky meet at FRB with Gerald Corrigan and FRB staff.
9:00 a.m.	Rosanne Witmer and Steven Jukofsky address representatives of the New York Clearing House banks assembled at the FRB for briefing on computer problem.
10:00 a.m.	Reconstruction and correction of Thursday customer data files completed.
12:15 p.m.	GSCS begins processing Friday business; $22.6 billion loan paid in full during the course of the day.

Board of Governors of the Federal Reserve System
OMB Number 7100-0036
Federal Deposit Insurance Corporation
OMB Number 3064-0052
Office of Comptroller of the Currency
OMB Number 1557-0081
Expires January 31, 1987

Federal Financial Institutions Examination Council

Consolidated Reports of Condition and Income for A Bank With Domestic and Foreign Offices–FFIEC 031

Report at the close of business September 30, 1985

This report is required by law: 12 U.S.C. §324 (State member banks); 12 U.S.C. §1817 (State nonmember banks); and 12 U.S.C. §161 (National banks).

This report form is to be filed by banks with branches and consolidated subsidiaries in U.S. territories and possessions, Edge or Agreement subsidiaries, foreign branches, consolidated foreign subsidiaries, or International Banking Facilities.

NOTE: The Reports of Condition and Income must be signed by an authorized officer and the Report of Condition must be attested to by not less than two directors for State nonmember banks and three directors for State member and National banks.

I, Robert E. Keilman, Senior Vice President & Comptroller
Name and Title of Officer Authorized to Sign Report

of the named bank do hereby declare that these Reports of Condition and Income (including the supporting schedules) have been prepared in conformance with the instructions issued by the appropriate Federal regulatory authority and are true to the best of my knowledge and belief.

Signature of Officer Authorized to Sign Report

October 29, 1985
Date of Signature

The Reports of Condition and Income are to be prepared in accordance with Federal Regulatory Authority instructions. NOTE: These instructions may in some cases differ from generally accepted accounting principles.

We, the undersigned directors, attest to the correctness of this Report of Condition (including the supporting schedules) and declare that it has been examined by us and to the best of our knowledge and belief has been prepared in conformance with the instructions issued by the appropriate Federal regulatory authority and is true and correct.

Director

Director

Director

State Member Banks: Return the original and one copy to the appropriate Federal Reserve District Bank.

State Nonmember Banks: Return the original only in the *special return address envelope provided.* If express mail is used in lieu of the special return address envelope, return the original only to the FDIC, Call Reports Analyses Unit, 550 17th Street, N.W., Washington, D.C. 20429.

National Banks: Return the original only in *the special return address envelope provided.* If express mail is used in lieu of the special return address envelope, return the original only to the FDIC, Call Reports Analyses Unit, 550 17th Street, N.W., Washington, D.C. 20429. Also send one copy to the appropriate District Deputy Comptroller and to the appropriate Federal Reserve District Bank.

Bank's Transit Number 0 2 1 0 0 0 0 1

State nonmember and National banks should affix the address label in this space.

The Bank of New York
Legal Title of Bank

New York — New York
City — County

New York — 10015
State — Zip Code

Board of Governors of the Federal Reserve System, Federal Deposit Insurance Corporation, Office of the Comptroller of the Currency

Consolidated Reports of Condition and Income for A Bank With Domestic and Foreign Offices

Table of Contents

Report of Condition

Schedule RC—Balance Sheet RC-1, 2
Schedule RC-A—Cash and Balances Due From Depository Institutions RC-3
Schedule RC-B—Securities RC-3
Schedule RC-C—Loans and Lease Financing Receivables RC-4
Schedule RC-D—Assets Held in Trading Accounts in Domestic Offices Only (to be completed only by banks with $1 billion or more in total assets) RC-5
Schedule RC-E—Deposit Liabilities RC-6, 7
Schedule RC-F—Other Assets RC-7
Schedule RC-G—Other Liabilities RC-7
Schedule RC-H—Selected Balance Sheet Items for Domestic Offices RC-8
Schedule RC-I—Selected Assets and Liabilities of IBFs RC-8
Schedule RC-J—Repricing Opportunities for Selected Balance Sheet Categories RC-9, 10, 11
Schedule RC-K—Quarterly Averages RC-12
Schedule RC-L—Commitments and Contingencies . RC-13
Schedule RC-M—Memoranda RC-14
Schedule RC-N—Past Due, Nonaccrual, and Renegotiated Loans and Lease Financing Receivables RC-15, 16
Schedule RC-O—Other Data for Deposit Insurance Assessments RC-17

Report of Income

Schedule RI—Income Statement RI-1, 2, 3
Schedule RI-A—Changes in Equity Capital RI-3
Schedule RI-B—Charge-Offs and Recoveries and Changes in Allowance for Loan and Lease Losses . RI-4
Schedule RI-C—Applicable Income Taxes by Taxing Authority RI-5
Schedule RI-D—Income from International Operations RI-6
Schedule RI-E—Explanations RI-7

The Bank of New York
Legal Title of Bank

New York, — City
New York — County
New York — State
10015 — Zip Code

Bank's Transit Number 0 2 1 0 0 0 0 1

Consolidated Report of Condition for Insured Commercial Banks for September 30, 1985

All schedules are to be reported in thousands of dollars. Unless otherwise indicated, report the amount outstanding as of the last business day of the quarter.

Schedule RC—Balance Sheet

C400 ◄

	Dollar Amounts in Thousands		Bil	Mil	Thou	
ASSETS						
1.	Cash and balances due from depository institutions (from Schedule RC-A):					
	a. Noninterest-bearing balances and currency and coin[1]	RCFD [illegible]		878	462	1.a.
	b. Interest-bearing balances	RCFD [illegible]		496	548	1.b.
2.	Securities (from Schedule RC-B)	RCFD [illegible]	1	514	406	2.
3.	Federal funds sold and securities purchased under agreements to resell in domestic offices of the bank and of its Edge and Agreement subsidiaries, and in IBFs	RCFD [illegible]	1	048	250	3.
4.	Loans and lease financing receivables:					
	a. Loans and leases, net of unearned income (from Schedule RC-C)	RCFD [illegible]	10	826	729	4.a.
	b. LESS: Allowance for loan and lease losses	RCFD [illegible]		159	000	4.b.
	c. LESS: Allocated transfer risk reserve	RCFD [illegible]		None		4.c.
	d. Loans and leases, net of unearned income, allowance, and reserve (item 4.a minus 4.b and 4.c)	RCFD [illegible]	10	667	729	4.d.
5.	Assets held in trading accounts	RCFD [illegible]		40	507	5.
6.	Premises and fixed assets (including capitalized leases)	RCFD [illegible]		119	297	6.
7.	Other real estate owned	RCFD [illegible]		18	483	7.
8.	Investments in unconsolidated subsidiaries and associated companies	RCFD [illegible]		16	220	8.
9.	Customers' liability to this bank on acceptances outstanding	RCFD [illegible]		647	460	9.
10.	Intangible assets (from Schedule RC-M)	RCFD 2143		1	768	10.
11.	Other assets (from Schedule RC-F)	RCFD [illegible]		415	448	11.
12.	Total assets (sum of items 1 through 11)	RCFD [illegible]	15	864	578	12.

---

[1] Includes cash items in process of collection and unposted debits.

Schedule RC—Continued

Dollar Amounts in Thousands		Bil	Mil	Thou	
LIABILITIES					
13. Deposits:					
a. In domestic offices (sum of totals of columns A and C from Schedule RC-E, part I)	RCON	7	870	663	13.a.
(1) Noninterest-bearing[1]	RCON	3	321	163	13.a.(1)
(2) Interest-bearing	RCON	4	549	480	13.a.(2)
b. In foreign offices, Edge and Agreement subsidiaries, and IBFs (from Schedule RC-E, part II)	RCFN	3	330	580	13.b.
(1) Noninterest-bearing	RCFN		7	933	13.b.(1)
(2) Interest-bearing	RCFN	3	322	647	13.b.(2)
14. Federal funds purchased and securities sold under agreements to repurchase in domestic offices of the bank and of its Edge and Agreement subsidiaries, and in IBFs	RCFD	2	275	634	14.
15. Demand notes issued to the U.S. Treasury	RCON		43	687	15.
16. Other borrowed money	RCFD		475	383	16.
17. Mortgage indebtedness and obligations under capitalized leases	RCFD 2910		2	240	17.
18. Bank's liability on acceptances executed and outstanding	RCFD		647	460	18.
19. Notes and debentures subordinated to deposits	RCFD		1	060	19.
20. Other liabilities (from Schedule RC-G)	RCFD		428	776	20.
21. Total liabilities (sum of items 13 through 20)	RCFD	15	075	483	21.
22. Limited-life preferred stock	RCFD		None		22.
EQUITY CAPITAL					
23. Perpetual preferred stock	RCFD		None		23.
24. Common stock	RCFD		112	272	24.
25. Surplus	RCFD		238	531	25.
26. Undivided profits and capital reserves	RCFD 3247		438	041	26.
27. Cumulative foreign currency translation adjustments	RCFD			251	27.
28. Total equity capital (sum of items 23 through 27)	RCFD 3210		789	095	28.
29. Total liabilities, limited-life preferred stock, and equity capital (sum of items 21, 22, and 28)	RCFD	15	864	578	29.

---

[1] Includes total demand deposits and noninterest-bearing time and savings deposits.

The Bank of New York
Legal Title of Bank

Schedule RC-A—Cash and Balances Due From Depository Institutions

Exclude assets held in trading accounts.

C405 ◄

Dollar Amounts in Thousands	(Column A) Consolidated Bank				(Column B) Domestic Offices				
		Bil	Mil	Thou		Bil	Mil	Thou	
1. Cash items in process of collection, unposted debits, and currency and coin	RCFD		745	142					1.
a. Cash items in process of collection and unposted debits					RCON		681	049	1.a.
b. Currency and coin					RCON		64	044	1.b.
2. Balances due from depository institutions in the U.S.					RCON		34	197	2.
a. U.S. branches and agencies of foreign banks (including their IBFs)	RCFD		68	500					2.a.
b. Other depository institutions in the U.S. (including their IBFs)	RCFD		39	458					2.b.
3. Balances due from banks in foreign countries and foreign central banks					RCON		6	947	3.
a. Foreign branches of other U.S. banks	RCFD		33	519					3.a.
b. Other banks in foreign countries and foreign central banks	RCFD		406	453					3.b.
4. Balances due from Federal Reserve Banks	RCFD		81	938	RCON		81	587	4.
5. Total (sum of items 1 through 4) (total of column A must equal Schedule RC, item 1)	RCFD	1	375	010	RCON		867	824	5.

Memorandum

1. Noninterest-bearing balances due from commercial banks in the U.S. (included in item 2, column B above)	RCON		34	197	M.1.

Schedule RC-B—Securities

Exclude assets held in trading accounts.

C410 ◄

Dollar Amounts in Thousands	Consolidated Bank								Domestic Offices				
	(Column A) Book Value				(Column B) Market Value[1]				(Column C) Book Value				
		Bil	Mil	Thou		Bil	Mil	Thou		Bil	Mil	Thou	
1. U.S. Treasury securities	RCFD		489	466	RCFD		502	579	RCON		489	466	1.
2. U.S. Government agency and corporation obligations:													
a. All holdings of U.S. Government-issued or -guaranteed certificates of participation in pools of residential mortgages	RCFD		2	665	RCFD		2	819	RCON		2	665	2.a.
b. All other	RCFD		38	012	RCFD		37	408	RCON		38	012	2.b.
3. Securities issued by states and political subdivisions in the U.S.	RCFD		963	414	RCFD		934	056	RCON		963	414	3.
4. Other domestic securities (debt and equity):													
a. All holdings of private (i.e., nongovernment-issued or -guaranteed) certificates of participation in pools of residential mortgages	RCFD		None		RCFD		None		RCON		None		4.a.
b. All other (includes Federal Reserve stock)	RCFD		10	009	RCFD		10	538	RCON		10	009	4.b.
5. Foreign securities (debt and equity)	RCFD		10	840	RCFD		10	887	RCON		None		5.
6. Total (sum of items 1 through 5) (total of column A must equal Schedule RC, item 2)	RCFD	1	514	406	RCFD	1	498	287	RCON	1	503	566	6.

Memorandum

1. Pledged securities	RCFD	1	172	859	M.1.

---

[1] See discussion in Glossary entry for "market value of securities."

Schedule RC-C—Loans and Lease Financing Receivables

Net of unearned income and before adjustment for allowance for loan and lease losses. Exclude assets held in trading accounts.

C415 ◄

Dollar Amounts in Thousands	(Column A) Consolidated Bank Bil	Mil	Thou	(Column B) Domestic Offices Bil	Mil	Thou	
1. Loans secured by real estate	1	654	284				1.
a. Construction and land development					644	291	1.a.
b. Secured by farmland (including farm residential and other improvements)						612	1.b.
c. Secured by 1-4 family residential properties					481	241	1.c.
d. Secured by multifamily (5 or more) residential properties					41	744	1.d.
e. Secured by nonfarm nonresidential properties					572	043	1.e.
2. Loans to depository institutions:							
a. To commercial banks in the U.S.					94	382	2.a.
(1) To U.S. branches and agencies of foreign banks		31	177				2.a.(1)
(2) To other commercial banks in the U.S.		91	263				2.a.(2)
b. To other depository institutions in the U.S.		30	585		30	585	2.b
c. To banks in foreign countries					55	521	2.c.
(1) To foreign branches of other U.S. banks		None					2.c.(1)
(2) To other banks in foreign countries		275	776				2.c.(2)
3. Loans to finance agricultural production and other loans to farmers			100			100	3.
4. Commercial and industrial loans:							
a. To U.S. addressees (domicile)	3	733	478	2	030	631	4.a.
b. To non-U.S. addressees (domicile)		624	695		1	617	4.b.
5. Acceptances of other banks:							
a. Of U.S. banks		None			None		5.a.
b. Of foreign banks		None			None		5.b.
6. Loans to individuals for household, family, and other personal expenditures (includes purchased paper)				1	012	110	6.
a. Credit cards and related plans (includes check credit and other revolving credit plans)		41	106				6.a.
b. Other (includes single payment and installment)		975	621				6.b.
7. Loans to foreign governments and official institutions (including foreign central banks)		556	876		7	811	7.
8. Obligations (other than securities) of states and political subdivisions in the U.S.:							
a. Nonrated industrial development obligations		151	571		151	571	8.a.
b. Other obligations (excluding securities)		None			None		8.b.
9. Other loans	2	416	832				9.
a. Loans for purchasing or carrying securities (secured and unsecured)				1	728	336	9.a.
b. All other loans					611	239	9.b.
10. Lease financing receivables (net of unearned income)					198	634	10.
a. Of U.S. addressees (domicile)		198	634				10.a.
b. Of non-U.S. addressees (domicile)		1	094				10.b.
11. LESS: Any unearned income on loans reflected in items 1-9 above		156	363		155	544	11.
12. Total loans and leases, net of unearned income (sum of items 1 through 10 minus item 11)(total of column A must equal Schedule RC, item 4.a)	10	826	729	7	606	924	12.
Memorandum							
1. Commercial paper included in Schedule RC-C above		None			None		M.1.

BEST COPY AVAILABLE

The Bank of New York
Legal Title of Bank

Schedule RC-D is to be completed only by banks with $1 billion or more in total assets.

Schedule RC-D—Assets Held in Trading Accounts in Domestic Offices Only

				C420	
		Domestic Offices			
Dollar Amounts in Thousands		Bil	Mil	Thou	
1. U.S. Treasury securities	RCON [illegible]		None		1.
2. U.S. Government agency and corporation obligations	RCON [illegible]		None		2.
3. Securities issued by states and political subdivisions in the U.S.	RCON [illegible]		40	507	3.
4. Other bonds, notes, and debentures	RCON [illegible]		None		4.
5. Certificates of deposit	RCON [illegible]		None		5.
6. Commercial paper	RCON [illegible]		None		6.
7. Banker's acceptances	RCON [illegible]		None		7.
8. Other	RCON [illegible]		None		8.
9. Total (sum of items 1 through 8)	RCON [illegible]		40	507	9.

Schedule RC-E—Deposit Liabilities

C425 ◄

Part I. Deposits in Domestic Offices

Dollar Amounts in Thousands	Transaction Accounts								Nontransaction Accounts				
	(Column A) Total transaction accounts (including total demand deposits)				(Column B) Memo: Total demand deposits (included in column A)				(Column C) Total nontransaction accounts (including MMDAs)				
		Bil	Mil	Thou		Bil	Mil	Thou		Bil	Mil	Thou	
Deposits of:													
1. Individuals, partnerships, and corporations	RCON	2	658	795	RCON	2	318	497	RCON	4	144	860	1.
2. U.S. Government	RCON		22	708	RCON		22	708	RCON		None		2.
3. States and political subdivisions in the U.S.	RCON		81	687	RCON		58	539	RCON		568	067	3.
4. Commercial banks in the U.S.	RCON		111	554	RCON		111	554					4.
a. U.S. branches and agencies of foreign banks									RCON		None		4.a.
b. Other commercial banks in the U.S.									RCON			104	4.b.
5. Other depository institutions in the U.S.	RCON		45	266	RCON		45	266	RCON		3	003	5.
6. Banks in foreign countries	RCON		53	351	RCON		53	351					6.
a. Foreign branches of other U.S. banks									RCON		None		6.a.
b. Other banks in foreign countries									RCON		None		6.b.
7. Foreign governments and official institutions (including foreign central banks)	RCON		1	959	RCON		1	959	RCON		30	000	7.
8. Certified and official checks	RCON		149	309	RCON		149	309					8.
9. Total (sum of items 1 through 8) (sum of columns A and C must equal Schedule RC, item 13.a)	RCON	3	124	629	RCON	2	761	183	RCON	4	746	034	9.

Dollar Amounts in Thousands		Bil	Mil	Thou	
Memoranda					
1. Selected components of total deposits (i.e., sum of item 9, columns A and C):					
a. Total Individual Retirement Accounts (IRAs) and Keogh Plan accounts	RCON		218	641	M.1.a.
b. Total brokered deposits	RCON		None		M.1.b.
c. Total brokered retail deposits (included in memorandum item 1.b above)					
(1) Issued in denominations of $100,000 or less	RCON		None		M.1.c.(1)
(2) Issued in denominations greater than $100,000 and participated out by the broker in shares of $100,000 or less	RCON		None		M.1.c.(2)
2. Components of total nontransaction accounts (sum of Memoranda items 2.a through 2.d must equal item 9, column C above):					
a. Nontransaction savings deposits (including MMDAs)	RCON	2	573	240	M.2.a.
b. Total time deposits of less than $100,000	RCON		543	302	M.2.b.
c. Time certificates of deposit of $100,000 or more	RCON		452	598	M.2.c.
d. Open-account time deposits of $100,000 or more	RCON	1	176	894	M.2.d.
3. "Super NOW" accounts	RCON		51	323	M.3.
4. Money market deposit accounts (MMDAs)	RCON	2	175	924	M.4.

Deposit Totals for FDIC Insurance Assessments

5. Total deposits (sum of item 9, column A and item 9, column C) (must equal Schedule RC, item 13.a)	RCON	7	870	663	M.5.
a. Total demand deposits (must equal item 9, column B)	RCON	2	761	183	M.5.a.
b. Total time and savings deposits (must equal item 9, column A plus column C minus column B)	RCON	5	109	480	M.5.b.

BEST COPY AVAILABLE

Schedule RC-E—Continued

Part II. Deposits in Foreign Offices (including Edge and Agreement subsidiaries and IBFs)

Dollar Amounts in Thousands		Bil	Mil	Thou	
Deposits of:					
1. Individuals, partnerships, and corporations	RCFN 2621	2	071	387	1.
2. U.S. banks (including IBFs and foreign branches of U.S. banks)	RCFN 2623		718	132	2.
3. Foreign banks (including U.S. branches and agencies of foreign banks, including their IBFs)	RCFN 2625		301	242	3.
4. Foreign governments and official institutions (including foreign central banks)	RCFN 2650		239	646	4.
5. Certified and official checks	RCFN 2330			124	5.
6. All other deposits	RCFN 2668			49	6.
7. Total (sum of items 1 through 6) (must equal Schedule RC, item 13.b)	RCFN 2200	3	330	580	7.

Schedule RC-F—Other Assets

							C430		◄
Dollar Amounts in Thousands						Bil	Mil	Thou	
1. Income earned, not collected on loans					RCFD 2164		130	932	1.
2. Net deferred income taxes[1] (if *debit balance*)					RCFD 2148		None		2.
3. Other (itemize amounts that exceed 25% of item 4 of this schedule)					RCFD 2168		284	516	3.
Accounts Receivable			179	629					
4. Total (sum of items 1 through 3) (must equal Schedule RC, item 11)					RCFD 2160		415	448	4.

Schedule RC-G—Other Liabilities

							C435		◄
Dollar Amounts in Thousands						Bil	Mil	Thou	
1. Expenses accrued and unpaid					RCFD 2932		168	968	1.
2. Net deferred income taxes[1] (if *credit balance*)					RCFD 3049		32	965	2.
3. Minority interest in consolidated subsidiaries					RCFD 3000		None		3.
4. Other (itemize amounts that exceed 25% of item 5 of this schedule)					RCFD 2938		226	843	4.
Accounts Payable			183	840					
5. Total (sum of items 1 through 4) (must equal Schedule RC, item 20)					RCFD 2930		428	776	5.

---

[1] See discussion of deferred income taxes in Glossary entry on "income taxes."

Schedule RC-H—Selected Balance Sheet Items for Domestic Offices

			C440		
	Domestic Offices				
Dollar Amounts in Thousands		Bil	Mil	Thou	
1. Customers' liability to this bank on acceptances outstanding	RCON [illegible]		373	088	1.
2. Bank's liability on acceptances executed and outstanding	RCON [illegible]		373	088	2.
3. Federal funds sold and securities purchased under agreements to resell	RCON [illegible]	1	048	250	3.
4. Federal funds purchased and securities sold under agreements to repurchase	RCON [illegible]	2	275	634	4.
5. Other borrowed money	RCON [illegible]		474	950	5.
EITHER					
6. Net due *from* own foreign offices, Edge and Agreement subsidiaries, and IBFs	RCON [illegible]		311	933	6.
OR					
7. Net due *to* own foreign offices, Edge and Agreement subsidiaries, and IBFs	RCON [illegible]		None		7.
8. Total assets (excludes net due from foreign offices, Edge and Agreement subsidiaries, and IBFs)	RCON [illegible]	11	808	432	8.
9. Total liabilities (excludes net due to foreign offices, Edge and Agreement subsidiaries, and IBFs)	RCON [illegible]	11	379	156	9.
Memorandum (*to be completed only by banks with IBFs and other "foreign" offices*)					
EITHER					
1. Net due *from* the IBF of the domestic offices of the reporting bank	RCON [illegible]		5	435	M.1.
OR					
2. Net due *to* the IBF of the domestic offices of the reporting bank	RCON [illegible]		None		M.2.

Schedule RC-I—Selected Assets and Liabilities of IBFs

To be completed only by banks with IBFs and other "foreign" offices.

			C445		
Dollar Amounts in Thousands		Bil	Mil	Thou	
1. Total IBF assets of the consolidated bank (component of Schedule RC, item 12)	RCFN [illegible]		661	598	1.
2. Total IBF loans and lease financing receivables (component of Schedule RC-C, item 12, column A)	RCFN [illegible]		565	595	2.
3. IBF commercial and industrial loans (component of Schedule RC-C, item 4, column A)	RCFN [illegible]		116	480	3.
4. Total IBF liabilities (component of Schedule RC, item 21)	RCFN [illegible]		433	208	4.
5. IBF deposit liabilities due to banks, including other IBFs (component of Schedule RC-E, part II, items 2 and 3)	RCFN [illegible]		130	616	5.
6. Other IBF deposit liabilities (component of Schedule RC-E, part II, items 1, 4, 5, and 6)	RCFN [illegible]		296	884	6.

The Bank of New York
Legal Title of Bank

The information required in this schedule represents the distribution across columns A through F of totals reported elsewhere in the Report of Condition (unless the bank's smallest foreign offices have been excluded). The *totals* being distributed (recorded in column G) *must* match the amounts reported for the corresponding items elsewhere in the Report of Condition (unless the bank's smallest foreign offices have been excluded). The required matching for each line item is described in the instructions.

Schedule RC-J—Repricing Opportunities for Selected Balance Sheet Categories

Dollar Amounts in Thousands

PART I. SELECTED ASSETS Part I excludes trading account assets and equity securities.	(Column A) Assets with immediately adjustable interest rates or original maturity of one day		Remaining Maturity if Fixed Rate Earliest Possible Repricing Interval if Floating Rate (Column B) 3 months or less, excluding amounts reported in column A		(Column C) Over 3 months through 6 months		(Column D) Over 6 months through 12 months		(Column E) Over 1 year through 5 years		(Column F) Over 5 years		C450 (Column G) Total amount to be distributed in columns A through F (see instructions)		
	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	
1. All loans and all lease financing receivables other than those in nonaccrual status[1]	6,986	442	1,279	808	427	911	242	662	1,166	110	580	793	10,683	720	1.
	RCFD 1425		RCFD 1426		RCFD 1427		RCFD 1428		RCFD 1429		RCFD 1431		RCFD 1479		
2. Debt securities (i.e., U.S. Treasury securities, U.S. Government agency obligations, state and local securities, and other bonds, notes, and debentures) . . .	15	387	100	267	120	366	458	930	218	938	600	518	1,514	406	2
	RCFD 1946		RCFD 1947		RCFD 1948		RCFD 1949		RCFD 1951		RCFD 1952		RCFD [illegible]		
3. Other interest-bearing assets (i.e., interest-bearing balances due from depository institutions, federal funds sold, and securities purchased under agreements to resell)	1,103	747	339	263	85	747	9	510	6	531	None		1,544	798	3.
	RCFD 1954		RCFD 1956		RCFD 1957		RCFD 1958		RCFD 1959		RCFD 1961		RCFD 1962		

---

[1] Amortizing real estate loans secured by 1 to 4 family residential properties in domestic offices and installment loans to individuals for household, family, and other personal expenditures (excluding credit cards and related plans) in domestic offices may be spread in item 1 of part I above by selecting one of two methods at the reporting bank's option: (1) by scheduled amortization (*the preferred method*), or (2) by final maturity for fixed rate loans and by earliest repricing opportunity for floating rate loans (*the alternative method*). The option may be exercised independently for such 1 to 4 family mortgages and for such consumer installment loans but must be followed consistently for all such loans within each category. When Method (2), the alternative method, is selected, memorandum item 2.a or 2.b, or both, must be reported, as appropriate.

For floating rate loans, if Method (1) is selected, the scheduled payments should be spread only to the earliest repricing opportunity, at which point the total remaining balance should be reported. If Method (2) is selected, the entire balance of such loans should be reported by the earliest repricing opportunity.

BEST COPY AVAILABLE

Schedule RC-J—Continued

Memoranda

	Mil	Thou	
1. Fixed rate debt securities with remaining maturity of over 10 years (included in item 2, column F above)	558	920	M.1.
	RCFD 1553		

2. Additional information required of banks that use Method (2), *the alternative method* (see footnote 1), for reporting certain multipayment loans in domestic offices in part I, item 1 above. [Banks that report amortizing 1–4 family real estate loans or consumer installment loans in item 1 of part I above on the basis of scheduled amortization (Method (1)) must enter N/A in 2.a or 2.b, or both, below, as appropriate.]

	(Column A) Estimated regularly scheduled payments received during the calendar quarter ending with the report date		(Column B) Dollar amount outstanding as of the report date for loans reported in part I, item 1, above, by using Method (2) (the alternative method)		
	Mil	Thou	Mil	Thou	
a. Amortizing real estate loans secured by 1 to 4 family residential properties in domestic offices	10	263			M.2.a.
	RCON 1432				
b. Installment loans to individuals for household, family, and other personal expenditures (excluding credit cards and related plans) in domestic offices	80	921	971	004	M.2.b.
	RCON 1433		RCON 1478		

The Bank of New York
Legal Title of Bank

Schedule RC-J—Continued

Dollar Amounts in Thousands

PART II. SELECTED LIABILITIES Part II excludes deposits in domestic offices classified as demand or savings accounts, such as money market deposit accounts.	(Column A) Liabilities with immediately adjustable interest rates or original maturity of one day		Remaining Maturity if Fixed Rate Earliest Possible Repricing Interval if Floating Rate										(Column G) Total amount to be distributed in columns A through F (see instructions)		
			(Column B) 3 months or less, excluding amounts reported in column A		(Column C) Over 3 months through 6 months		(Column D) Over 6 months through 12 months		(Column E) Over 1 year through 5 years		(Column F) Over 5 years				
	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	Mil	Thou	
1. Time deposits in domestic offices:															
a. Time certificates of deposit of $100,000 or more (excluding open-account time deposits of $100,000 or more)	25	615	338	346	42	953	26	684	19	000	None		452	598	1.a
	RCON 2458		RCON 2460		RCON 2461		RCON 2462		RCON 2464		RCON 2465		RCON 6645		
b. All other time deposits (including all open-account time deposits)	49	649	1,262	405	222	158	96	130	89	848		6	1,720	196	1.b.
	RCON 2466		RCON 2467		RCON 2470		RCON 2471		RCON 2472		RCON 2473		RCON 2477		
2. Total deposits in foreign offices, Edge and Agreement subsidiaries, and IBFs	554	488	2,186	356	523	252	51	554	14	930	None		3,330	580	2.
	RCFN 2474		RCFN 2475		RCFN 2846		RCFN 2847		RCFN 2848		RCFN 2849		RCFN 2100		
3. Nondeposit interest-bearing liabilities (i.e., federal funds purchased, securities sold under agreements to repurchase, other borrowed money, and notes and debentures subordinated to deposits)	2,346	067	261	650	107	300	16	000	1	060	20	000	2,752	077	3.
	RCFD 2482		RCFD 2483		RCFD 2485		RCFD 2487		RCFD 2489		RCFD 2501		RCFD 2527		

Schedule RC-K—Quarterly Averages[1]

Dollar Amounts in Thousands		Bil	Mil	Thou	C455
ASSETS					
1. Interest-bearing balances due from depository institutions	RCFD 3381		706	372	1.
2. U.S. Treasury securities and U.S. Government agency and corporation obligations	RCFD [illegible]		530	858	2.
3. Securities issued by states and political subdivisions in the U.S.	RCFD [illegible]		915	603	3.
4. Other securities (debt and equity)	RCFD [illegible]		22	099	4.
5. Federal funds sold and securities purchased under agreements to resell in domestic offices of the bank and of its Edge and Agreement subsidiaries, and in IBFs	RCFD [illegible]		883	047	5.
6. Loans:					
a. Loans in domestic offices:					
(1) Total loans	RCON [illegible]	6	205	123	6.a.(1)
(2) Loans secured by real estate	RCON [illegible]	1	861	692	6.a.(2)
(3) Loans to finance agricultural production and other loans to farmers[2]	RCON [illegible]		None		6.a.(3)
(4) Commercial and industrial loans	RCON [illegible]	1	727	762	6.a.(4)
(5) Loans to individuals for household, family, and other personal expenditures	RCON [illegible]	1	003	784	6.a.(5)
(6) Obligations (other than securities) of states and political subdivisions in the U.S.	RCON [illegible]		145	263	6.a.(6)
b. Total loans in foreign offices, Edge and Agreement subsidiaries, and IBFs	RCFN [illegible]	3	183	823	6.b.
7. Assets held in trading accounts	RCFD 3401		46	807	7.
8. Earning assets	RCFD 3402	12	638	445	8.
9. Total assets	RCFD [illegible]	14	601	284	9.
LIABILITIES					
10. Time certificates of deposit of $100,000 or more in domestic offices	RCON [illegible]		438	916	10.
11. "Super NOW" accounts, money market deposit accounts, and time deposits (other than certificates of deposit of $100,000 or more) in domestic offices	RCON [illegible]	3	383	466	11.
12. Interest-bearing deposits in foreign offices, Edge and Agreement subsidiaries, and IBFs	RCFN [illegible]	3	149	654	12.
13. Federal funds purchased and securities sold under agreements to repurchase in domestic offices of the bank and of its Edge and Agreement subsidiaries, and in IBFs	RCFD [illegible]	1	660	578	13.
14. Other borrowed money	RCFD [illegible]		568	310	14.

---

[1] For all items, banks have the option of reporting either (1) an average of daily figures for the quarter, or (2) an average of weekly figures (i.e., the Wednesday of each week of the quarter).

[2] Must be reported separately only if agricultural loans in domestic offices exceed five percent of total loans in domestic offices.

BEST COPY AVAILABLE

The Bank of New York
Legal Title of Bank

Bank's Transit Number 0 2 1 0 0 0 0 1

Schedule RC-L—Commitments and Contingencies

Please read carefully the instructions for the preparation of Schedule RC-L.

Dollar Amounts in Thousands		Bil	Mil	Thou	
			C460		
1. Commitments to make or purchase loans or to extend credit in the form of lease financing arrangements (report only the unused portions of commitments that are fee paid or otherwise legally binding)	RCFD 3423	4	560	609	1.
2. Futures and forward contracts (exclude contracts involving foreign exchange):					
a. Commitments to purchase	RCFD 3424		15	380	2.a.
b. Commitments to sell	RCFD 3425		63	710	2.b.
3. When-issued securities:					
a. Gross commitments to purchase	RCFD 3434		3	670	3.a.
b. Gross commitments to sell	RCFD 3435			120	3.b.
4. Standby contracts and other option arrangements:					
a. Obligations to purchase under option contracts	RCFD 3426		None		4.a.
b. Obligations to sell under option contracts	RCFD 3427		None		4.b.
5. Commitments to purchase foreign currencies and U.S. dollar exchange (spot and forward)	RCFD 3415	5	650	286	5.
6. Standby letters of credit and foreign office guarantees:					
a. Standby letters of credit and foreign office guarantees:					
(1) To U.S. addressees (domicile)	RCFD 3376	1	220	435	6.a.(1)
(2) To non-U.S. addressees (domicile)	RCFD 3377		147	435	6.a.(2)
b. Amount of standby letters of credit in items 6.a.(1) and 6.a.(2) conveyed to others through participations	RCFD 3378		148	119	6.b.
7. Commercial and similar letters of credit	RCFD 3411		161	407	7.
8. Participations in acceptances (as described in the instructions) conveyed to others by the reporting (accepting) bank	RCFD 3428		None		8.
9. Participations in acceptances (as described in the instructions) acquired by the reporting (nonaccepting) bank	RCFD 3429		None		9.
10. Securities borrowed	RCFD 3432		None		10.
11. Securities lent	RCFD 3433		None		11.
12. Other significant commitments and contingencies (list below each component of this item over 25% of Schedule RC, item 28, "Total equity capital")	RCFD 3430		221	682	12.

Memoranda

		Bil	Mil	Thou	
1. Loans originated by the reporting bank that have been sold or participated to others during the calendar quarter ending with the report date (exclude the portions of such loans retained by the reporting bank; see instructions for other exclusions)	RCFD 3431		937	561	M.1.
2. Notional value of all outstanding interest rate swaps	RCFD 3450		861	876	M.2.

BEST COPY AVAILABLE

Bank's Transit Number 0 2 1 0 0 0 0 1

The Bank of New York
Legal Title of Bank

Schedule RC-M—Memoranda

			C465		
Dollar Amounts in Thousands		Bil	Mil	Thou	
1. Extensions of credit by the reporting bank to its executive officers, principal shareholders, and their related interests as of the report date:					
a. Aggregate amount of all extensions of credit to all executive officers, principal shareholders, and their related interests	RCFD 6164			82	1.a.
b. Number of executive officers and principal shareholders to whom the amount of all extensions of credit by the reporting bank (including extensions of credit to related interests) equals or exceeds the lesser of $500,000 or 5 percent of total capital as defined for this purpose in agency regulations — Number: RCFD [illegible] None					1.b.
2. Federal funds sold and securities purchased under agreements to resell (Schedule RC, item 3) with U.S. branches and agencies of foreign banks	RCFD 3405		255	250	2.
3. Amount of reserve balances actually passed through to the Federal Reserve by the reporting bank on behalf of its respondent depository institutions[1]	RCFD [illegible]		1	193	3.
4. Amount of reserve balances actually passed through to the Federal Reserve on behalf of the reporting bank by its correspondent bank[1]	RCFD [illegible]		None		4.
5. To be completed only by banks with $1 billion or more in total assets: Customers' liability to this bank on acceptaces outstanding (sum of items 5.a and 5.b must equal Schedule RC, item 9):					
a. U.S. addressees (domicile)	RCFD 2103		95	761	5.a.
b. Non-U.S. addressees (domicile)	RCFD 2104		551	699	5.b.
6. Intangible assets:					
a. Mortgage servicing rights	RCFD 3164		None		6.a.
b. Other identifiable intangible assets	RCFD 3165		1	768	6.b.
c. Goodwill	RCFD 3163		None		6.c.
d. Total (sum of items 6.a through 6.c)(must equal Schedule RC, item 10)	RCFD 2143		1	768	6.d.
e. *To be completed by national banks only:* Unamortized balance of intangible assets booked prior to April 15, 1985 (included in item 6.d above)	RCFD [illegible]		None		6.e.
		YES		NO	
7. Does your bank have any mandatory convertible debt that is part of your primary or secondary capital?	RCFD [illegible]			X	7.
If yes, continue below:		Bil	Mil	Thou	
a. Total equity contract notes, gross	RCFD 3290				7.a.
b. Common or perpetual preferred stock dedicated to redeem the above notes	RCFD 3291				7.b.
c. Total equity commitment notes, gross	RCFD 3293				7.c.
d. Common or perpetual preferred stock dedicated to redeem the above notes	RCFD 3294				7.d.
e. Total (item 7.a minus 7.b plus 7.c minus 7.d)	RCFD 3295				7.e.

---

[1] See Glossary entry for "pass-through reserve balances."

The Bank of New York
Legal Title of Bank

Bank's Transit Number 0 2 1 0 0 0 0 1

Schedule RC-N—Past Due, Nonaccrual, and Renegotiated Loans and Lease Financing Receivables

The FFIEC regards only information reported in Column A confidential.

C470 ◄

	(Column A) Past due 30 through 89 days and still accruing				(Column B) Past due 90 days or more and still accruing				(Column C) Nonaccrual				(Column D) Renegotiated "troubled" debt				
Dollar Amounts in Thousands		Bil	Mil	Thou		Bil	Mil	Thou		Bil	Mil	Thou		Bil	Mil	Thou	
1. In domestic offices:																	
a. Loans secured by real estate	RCON 1421				RCON 1422		49	436	RCON [illegible]		121	851	RCON 1424			884	1.a.
b. Commercial and industrial loans	RCON [illegible]				RCON [illegible]		16	647	RCON [illegible]		117	820	RCON [illegible]		None		1.b.
c. Loans to individuals for household, family, and other personal expenditures	RCON [illegible]				RCON [illegible]		7	960	RCON [illegible]		6	674					1.c.
d. Loans to finance agricultural production and other loans to farmers[1]	RCON [illegible]				RCON [illegible]		None		RCON [illegible]		None		RCON [illegible]		None		1.d.
e. All other loans and all lease financing receivables in domestic offices	RCON [illegible]				RCON [illegible]		6	000	RCON [illegible]		23	747	RCON [illegible]		None		1.e.
2. In foreign offices, Edge and Agreement subsidiaries, and IBFs	RCFN [illegible]				RCFN [illegible]			97	RCFN [illegible]		29	274	RCFN [illegible]		None		2.
3. Total (sum of items 1.a through 1.e and 2)	RCFD [illegible]				RCFD [illegible]		80	140	RCFD [illegible]		299	366	RCFD [illegible]			884	3.

---

[1] Must be reported separately only if agricultural loans in domestic offices exceed five percent of total loans in domestic offices. If less than or equal to five percent, may be included in item 1.e.

BEST COPY AVAILABLE

Bank's Transit Number 0 2 1 0 0 0 0 1

FFIEC 031
Page RC-16

Optional Submission of Narrative Statement Concerning the Figures Submitted on Schedule RC-N, Past Due, Nonaccrual, and Renegotiated Loans and Lease Financing Receivables

at close of business on September 30, 19 85

The Bank of New York — Legal Title of Bank
New York — City
New York — State

The management of the reporting bank may, *if it wishes*, submit a brief narrative statement on the figures submitted on this schedule. This optional statement will be made available to the public, along with columns B, C, and D of this schedule and the Reports of Condition and Income, in response to any request for individual bank Report of Condition data or for this schedule. Banks choosing to submit the narrative statement should therefore ensure that the statement does not contain information that they are not willing to have made public or that would compromise the privacy of their customers.

The optional statement must be entered on this sheet. The statement should not exceed 100 words. Further, regardless of the number of words, the statement must not exceed 750 characters, including punctuation, indentation, and standard spacing between words and sentences. If any submission should exceed 750 characters, as defined, it will be truncated at 750 characters with no notice to the submitting bank and the truncated statement will appear as the bank's statement both on agency computerized records and in computer-file releases to the public.

All information furnished by the bank in the narrative statement must be accurate and not misleading. Appropriate efforts shall be taken by the submitting bank to ensure the statement's accuracy. The statement must be signed, in the space provided below, by a senior officer of the bank who thereby attests to its accuracy.

If, subsequent to the original submission, *material* changes are submitted for the data reported on this schedule, the existing narrative statement will be deleted from the files, and from disclosure; the bank, at its option, may replace it with a statement, under signature, appropriate to the amended data.

The optional narrative statement will appear in agency records and in release to the public exactly as submitted (or amended as described in the preceding paragraph) by the management of the bank (except for the truncation of overlength statements described above). THE STATEMENT WILL *NOT* BE EDITED OR SCREENED IN ANY WAY BY THE SUPERVISORY AGENCIES FOR ACCURACY OR RELEVANCE. DISCLOSURE OF THE STATEMENT SHALL NOT SIGNIFY THAT ANY FEDERAL SUPERVISORY AGENCY HAS VERIFIED OR CONFIRMED THE ACCURACY OF THE INFORMATION CONTAINED THEREIN. A STATEMENT TO THIS EFFECT WILL APPEAR ON ANY PUBLIC RELEASE OF THE OPTIONAL STATEMENT SUBMITTED BY THE MANAGEMENT OF THE REPORTING BANK.

Bank Management Statement (please type or print clearly): C471 ◄

Robert E. Keilman — 10/31/85
Signature of Executive Officer of Bank — Date of Signature
Robert E. Keilman, Senior Vice President
and Comptroller

BEST COPY AVAILABLE

Bank's Transit Number 0 2 1 0 0 0 0 1

FFIEC 031
Page RC-17

The Bank of New York
Legal Title of Bank

Schedule RC-O—Other Data for Deposit Insurance Assessments

Dollar Amounts in Thousands		Bil	Mil	Thou	
			C475		◄
1. Unposted debits (see instructions):					
a. Actual amount of all unposted debits	RCON 0030		None		1.a.
OR					
b. Separate amount of unposted debits:					
(1) Actual amount of unposted debits to demand deposits	RCON 0031		None		1.b.(1)
(2) Actual amount of unposted debits to time and savings deposits	RCON 0032		None		1.b.(2)
2. Unposted credits (see instructions):					
a. Actual amount of all unposted credits	RCON 3510		None		2.a.
OR					
b. Separate amount of unposted credits:					
(1) Actual amount of unposted credits to demand deposits	RCON 3512		None		2.b.(1)
(2) Actual amount of unposted credits to time and savings deposits	RCON 3514		None		2.b.(2)
3. Uninvested trust funds (cash) held in bank's own trust department (not included in total deposits in domestic offices)	RCON 3520		None		3.
4. Deposits of consolidated subsidiaries in domestic offices (not included in total deposits in domestic offices):					
a. Demand deposits of consolidated subsidiaries	RCON 2211			104	4.a.
b. Time and savings deposits of consolidated subsidiaries	RCON 2351		None		4.b.
5. Deposits of insured branches in Puerto Rico and U.S. territories and possessions (included in Schedule RC-E, part II):					
a. Demand deposits in insured branches	RCON 2229		None		5.a.
b. Time and savings deposits in insured branches	RCON 2383		None		5.b.

Item 6 is not applicable to state nonmember banks that have not been authorized by the Federal Reserve to act as pass-through correspondents.

		Bil	Mil	Thou	
6. Reserve balances actually passed through to the Federal Reserve by the reporting bank on behalf of its respondent depository institutions that are also reflected as deposit liabilities of the reporting bank:					
a. Amount reflected in demand deposits (included in Schedule RC-E, Part I, Memorandum item 5.a)	RCON 2314		1	193	6.a.
b. Amount reflected in time and savings deposits (included in Schedule RC-E, Part I, Memorandum item 5.b)	RCON 2315		None		6.b.

Memorandum

	Number		Bil	Mil	Thou	
1. Total deposits in domestic offices of the bank (to be completed for the *June* report only):						
a. Amount of deposit accounts $100,000 or less		RCON 2702				M.1.a.
b. Deposit accounts more than $100,000:						
(1) *Amount* outstanding		RCON 2710				M.1.b.(1)
(2) *Number* of accounts	RCON 2722					M.1.b.(2)

C477 ◄

Person to whom questions about the Report of Condition should be directed:

Name: Richard F. Miesemer
Title: Vice President
Area code and phone number: (212) 530-1125

Person(s) to whom questions about Schedules RC-J, L, and N should be directed if different than above:

Name ______ Title ______ Area code and phone number ______

Name ______ Title ______ Area code and phone number ______

BEST COPY AVAILABLE

The Bank of New York
Legal Title of Bank

New York	New York
City	County
New York	10015
State	Zip Code

Bank's Transit Number 0 2 1 0 0 0 0 1

Consolidated Report of Income
for the period January 1, 1985-September 30, 1985

All Report of Income schedules are to be reported on a calendar year-to-date basis in thousands of dollars

Schedule RI—Income Statement

			1480		
Dollar Amounts in Thousands		Bil	Mil	Thou	
1. Interest income:					
a. Interest and fee income on loans:					
(1) In domestic offices:					
(a) Loans secured by real estate	RIAD 4011		135	946	1.a.(1)(a)
(b) Loans to depository institutions	RIAD 4019		4	303	1.a.(1)(b)
(c) Loans to finance agricultural production and other loans to farmers	RIAD [illegible]			8	1.a.(1)(c)
(d) Commercial and industrial loans	RIAD [illegible]		141	215	1.a.(1)(d)
(e) Acceptances of other banks	RIAD [illegible]		None		1.a.(1)(e)
(f) Loans to individuals for household, family, and other personal expenditures:					
(1) Credit cards and related plans	RIAD [illegible]		5	315	1.a.(1)(f)(1)
(2) Other	RIAD [illegible]		72	164	1.a.(1)(f)(2)
(g) Loans to foreign governments and official institutions	RIAD [illegible]			719	1.a.(1)(g)
(h) Obligations (other than securities) of states and political subdivisions in the U.S.	RIAD [illegible]		8	291	1.a.(1)(h)
(i) All other loans in domestic offices	RIAD [illegible]		85	003	1.a.(1)(i)
(2) In foreign offices, Edge and Agreement subsidiaries, and IBFs	RIAD [illegible]		231	811	1.a.(2)
b. Income from lease financing receivables	RIAD [illegible]		4	413	1.b.
c. Interest income on balances due from depository institutions:					
(1) In domestic offices	RIAD [illegible]		None		1.c.(1)
(2) In foreign offices, Edge and Agreement subsidiaries, and IBFs	RIAD [illegible]		78	011	1.c.(2)
d. Interest and dividend income on securities:					
(1) U.S. Treasury securities and U.S. Government agency and corporation obligations	RIAD [illegible]		44	376	1.d.(1)
(2) Securities issued by states and political subdivisions in the U.S.	RIAD [illegible]		45	847	1.d.(2)
(3) Other domestic securities (debt and equity)	RIAD [illegible]			491	1.d.(3)
(4) Foreign securities (debt and equity)	RIAD [illegible]			784	1.d.(4)
e. Interest income from assets held in trading accounts	RIAD [illegible]		1	897	1.e.
f. Interest income on federal funds sold and securities purchased under agreements to resell in domestic offices of the bank and of its Edge and Agreement subsidiaries, and in IBFs	RIAD [illegible]		57	803	1.f.
g. Total interest income (sum of items 1.a through 1.f)	RIAD 4107		918	397	1.g.

Schedule RI—Continued

Dollar Amounts in Thousands		Bil	Mil	Thou			Bil	Mil	Thou	
2. Interest expense:										
a. Interest on deposits:										
(1) Interest on deposits in domestic offices:										
(a) Interest on time certificates of deposit of $100,000 or more	RIAD 4174		28	794	2.a.(1)(a)					
(b) Interest on other deposits	RIAD 4176		215	838	2.a.(1)(b)					
(2) Interest on deposits in foreign offices, Edge and Agreement subsidiaries, and IBFs	RIAD 4172		228	337	2.a.(2)					
b. Expense of federal funds purchased and securities sold under agreements to repurchase in domestic offices of the bank and of its Edge and Agreement subsidiaries, and in IBFs	RIAD 4180		98	648	2.b.					
c. Interest on demand notes issued to the U.S. Treasury and on other borrowed money	RIAD 4185		36	680	2.c.					
d. Interest on mortgage indebtedness and obligations under capitalized leases	RIAD 4072			192	2.d.					
e. Interest on notes and debentures subordinated to deposits	RIAD 4200			256	2.e.					
f. Total interest expense (sum of items 2.a through 2.e)	RIAD 4073		608	745	2.f.					
3. Net interest income (item 1.g minus 2.f)						RIAD 4074		309	652	3.
4. Provisions:										
a. Provision for loan and lease losses						RIAD 4230		81	173	4.a.
b. Provision for allocated transfer risk						RIAD 4243		None		4.b.
5. Noninterest income:										
a. Income from fiduciary activities	RIAD 4070		122	127	5.a.					
b. Service charges on deposit accounts in domestic offices	RIAD 4080		18	142	5.b.					
c. Trading gains (losses) and fees from foreign exchange transactions	RIAD 4075		4	742	5.c.					
d. Other foreign transaction gains (losses)	RIAD 4076		—	(39)	5.d.					
e. Gains (losses) and fees from assets held in trading accounts	RIAD 4077			316	5.e.					
f. Other noninterest income	RIAD 4078		28	950	5.f.					
g. Total noninterest income (sum of items 5.a through 5.f)						RIAD 4079		174	238	5.g.
6. Gains (losses) on securities not held in trading accounts						RIAD 4091		(1	791)	6.
7. Noninterest expense:										
a. Salaries and employee benefits	RIAD 4135		157	332	7.a.					
b. Expenses of premises and fixed assets (net of rental income) (excluding salaries and employee benefits and mortgage interest)	RIAD 4217		61	667	7.b.					
c. Other noninterest expense	RIAD 4092		78	706	7.c.					
d. Total noninterest expense (sum of items 7.a through 7.c)						RIAD 4093		297	705	7.d.
8. Income (loss) before income taxes and extraordinary items and other adjustments (item 3 plus or minus items 4.a, 4.b, 5.g, 6, and 7.d)						RIAD 4301		103	221	8.
9. Applicable income taxes (on item 8)						RIAD 4302		28	687	9.
10. Income (loss) before extraordinary items and other adjustments (item 8 minus 9)						RIAD 4300		74	534	10.
11. Extraordinary items and other adjustments:										
a. Extraordinary items and other adjustments, gross of income taxes	RIAD 4310		None		11.a.					
b. Applicable income taxes (on item 11.a)	RIAD 4315		None		11.b.					
c. Extraordinary items and other adjustments, net of income taxes (item 11.a minus 11.b)						RIAD 4320		None		11.c.
12. Net income (loss) (sum of items 10 and 11.c)						RIAD 4340		74	534	12.

The Bank of New York
Legal Title of Bank

Schedule RI—Continued

Dollar Amounts in Thousands	Year-to-date Bil	Year-to-date Mil	Year-to-date Thou	
Memoranda				
1. Interest on "Super NOW" accounts, money market deposit accounts, and time deposits (other than certificates of deposit of $100,000 or more) in domestic offices		189	061	M.1.
2. Estimated tax-exempt lease income		None		M.2.
3. Estimated investment tax credit		10	650	M.3.
4. Estimated foreign tax credit included in applicable income taxes, items 9 and 11.b.		1	500	M.4.
5. Income taxes applicable to gains (losses) on securities not held in trading accounts (included in item 9 above)			(884)	M.5.

	Number	
6. Number of full-time equivalent employees on payroll at end of current period	7,478	M.6.

Schedule RI-A—Changes in Equity Capital

Indicate decreases and losses in parentheses.

Dollar Amounts in Thousands	Bil	Mil	Thou	
		1483		
1. Equity capital end of previous calendar year		680	910	1.
2. Equity capital adjustments from amended Reports of Income, net		None		2.
3. Amended balance end of previous calendar year (sum of items 1 and 2)		680	910	3.
4. Net income (loss) (must equal Schedule RI, item 12)		74	534	4.
5. Sale, conversion, acquisition, or retirement of capital stock, net		None		5.
6. Changes incident to business combinations, net		None		6.
7. LESS: Cash dividends declared on preferred stock		None		7.
8. LESS: Cash dividends declared on common stock		46	600	8.
9. Cumulative effect of changes in accounting principles from prior years (see instructions for this schedule)		None		9.
10. Corrections of material accounting errors from prior years (see instructions for this schedule)		None		10.
11. Change in net unrealized loss on marketable equity securities		None		11.
12. Foreign currency translation adjustments			251	12.
13. Other transactions with parent holding company (not included in items 5, 7, or 8 above)		80	000	13.
14. Equity capital end of current period (sum of items 3 through 13) (must equal Schedule RC, item 28)		789	095	14.

Schedule RI-B—Charge-Offs and Recoveries and Changes in Allowance for Loan and Lease Losses

Part I. Charge-Offs and Recoveries on Loans and Leases

Part I excludes charge-offs and recoveries through the allocated transfer risk reserve.

1486 ◄

Dollar Amounts in Thousands		(Column A) Charge-Offs calendar year-to-date				(Column B) Recoveries calendar year-to-date			
		Bil	Mil	Thou		Bil	Mil	Thou	
1. Loans secured by real estate:									
a. To U.S. addressees (domicile)	RIAD 4651		1	424	RIAD 4661			970	1.a.
b. To non-U.S. addressees (domicile)	RIAD 4652		None		RIAD 4662		None		1.b.
2. Loans to depository institutions and acceptances of other banks:									
a. To U.S. banks and other U.S. depository institutions	RIAD 4653		None		RIAD 4663		None		2.a.
b. To foreign banks	RIAD 4654		5	000	RIAD 4664			29	2.b.
3. Loans to finance agricultural production and other loans to farmers	RIAD 4655		None		RIAD 4665		None		3.
4. Commercial and industrial loans:									
a. To U.S. addressees (domicile)	RIAD 4645		26	620	RIAD 4617		2	887	4.a.
b. To non-U.S. addressees (domicile)	RIAD 4646		5	500	RIAD 4618		None		4.b.
5. Loans to individuals for household, family, and other personal expenditures:									
a. Credit cards and related plans	RIAD 4656		1	147	RIAD 4666			870	5.a.
b. Other	RIAD 4657		3	675	RIAD 4667		1	234	5.b.
6. Loans to foreign governments and official institutions	RIAD 4643		13	854	RIAD 4627		None		6.
7. All other loans	RIAD 4644			258	RIAD 4628			45	7.
8. Lease financing receivables:									
a. Of U.S. addressees (domicile)	RIAD 4658		None		RIAD 4668		None		8.a.
b. Of non-U.S. addressees (domicile)	RIAD 4659		None		RIAD 4669		None		8.b.
9. Total (sum of items 1 through 8)	RIAD 4635		57	478	RIAD 4605		6	035	9.

Part II. Changes in Allowance for Loan and Lease Losses and in Allocated Transfer Risk Reserve

Dollar Amounts in Thousands		(Column A) Allowance for Loan and Lease Losses				(Column B) Allocated Transfer Risk Reserve			
		Bil	Mil	Thou		Bil	Mil	Thou	
1. Balance end of previous calendar year	RIAD 3124		129	270	RIAD 3131		None		1.
2. Recoveries (column A must equal part I, item 9, column B above)	RIAD 4605		6	035	RIAD 3132		None		2.
3. LESS: Charge-offs (column A must equal part I, item 9, column A above)	RIAD 4635		57	478	RIAD 3133		None		3.
4. Provision (column A must equal Schedule RI, item 4.a; column B must equal Schedule RI, item 4.b)	RIAD 4230		81	173	RIAD 4243		None		4.
5. Adjustments (see instructions for this schedule)	RIAD 4815		None		RIAD 3134		None		5.
6. Balance end of current period (sum of items 1 through 5) (column A must equal Schedule RC, item 4.b; column B must equal Schedule RC, item 4.c)	RIAD 3123		159	000	RIAD 3128		None		6.

BEST COPY AVAILABLE

Schedule RI-D—Income from International Operations

For all banks with foreign offices, Edge or Agreement subsidiaries, or IBFs where international operations account for more than 10 percent of total revenues, total assets, or net income.

Part I. Estimated Income from International Operations

Dollar Amounts in Thousands		Bil	1492 Mil	 Thou	◄
1. Interest income and expense booked at foreign offices, Edge and Agreement subsidiaries, and IBFs:					
a. Interest income booked	RIAD [illegible]		312	404	1.a.
b. Interest expense booked	RIAD [illegible]		266	469	1.b.
c. Net interest income booked at foreign offices, Edge and Agreement subsidiaries, and IBFs (item 1.a minus 1.b)	RIAD [illegible]		45	935	1.c.
2. Adjustments for booking location of international operations:					
a. Net interest income attributable to international operations booked at domestic offices	RIAD [illegible]		23	686	2.a.
b. Net interest income attributable to domestic business booked at foreign offices	RIAD [illegible]		40	656	2.b.
c. Net booking location adjustment (item 2.a minus 2.b)	RIAD [illegible]		(16	970)	2.c.
3. Noninterest income and expense attributable to international operations:					
a. Noninterest income attributable to international operations	RIAD [illegible]		10	387	3.a.
b. Provision for loan and lease losses attributable to international operations	RIAD [illegible]		24	354	3.b.
c. Other noninterest expense attributable to international operations	RIAD [illegible]		8	496	3.c.
d. Net noninterest income (expense) attributable to international operations (item 3.a minus 3.b and 3.c)	RIAD [illegible]		(22	463)	3.d.
4. Estimated pretax income attributable to international operations before capital allocation adjustment (sum of items 1.c, 2.c, and 3.d)	RIAD [illegible]		6	502	4.
5. Adjustment to pretax income for internal allocations to international operations to reflect the effects of equity capital on overall bank funding costs	RIAD [illegible]		(9	603)	5.
6. Estimated pretax income attributable to international operations after capital allocation adjustment (sum of items 4 and 5)	RIAD [illegible]		(3	101)	6.
7. Income taxes attributable to income from international operations as estimated in item 6	RIAD [illegible]		(1	426)	7.
8. Estimated net income attributable to international operations (item 6 minus 7)	RIAD [illegible]		(1	675)	8.

Memoranda

1. Intracompany interest income included in item 1.a above	RIAD [illegible]		1	799	M.1.
2. Intracompany interest expense included in item 1.b above	RIAD [illegible]		34	547	M.2.

Part II. Supplementary Details on Income from International Operations Required by the Departments of Commerce and Treasury for Purposes of the U.S. International Accounts and the U.S. National Income and Product Accounts

Dollar Amounts in Thousands		Bil	Mil	Thou	
1. Interest income booked at IBFs	RIAD [illegible]		50	058	1.
2. Interest expense booked at IBFs	RIAD [illegible]		37	776	2.
3. Noninterest income attributable to international operations booked at domestic offices (excluding IBFs)	RIAD [illegible]		7	643	3.
4. Provision for loan and lease losses attributable to international operations booked at domestic offices (excluding IBFs)	RIAD [illegible]		None		4.
5. Other noninterest expense attributable to international operations booked at domestic offices (excluding IBFs)	RIAD [illegible]			407	5.

BEST COPY AVAILABLE

The. Bank of New York
Legal Title of Bank

Schedule RI-E—Explanations

Detail all adjustments in Schedules RI-A and RI-B, all extraordinary items and other adjustments in Schedule RI, and all significant items of other noninterest income and other noninterest expense in Schedule RI. (See instructions for details.)

Schedule RI Item 5f includes international commissions $9,131M.

Schedule RI Item 7c includes amortization expense of intangible assets $308M.

Person to whom questions about the Report of Income should be directed:

Richard F. Miesemer	Vice President	(212) 530-1125
Name	Title	Area code and phone number

THE BANK OF NEW YORK (New York, N.Y.)

(Controlled by The Bank of New York Co., Inc.)

Established in 1784 under the leadership of Alexander Hamilton. Chartered in 1791. Bank was the first incorporated institution of its kind in New York. In 1922 merged with New York Life Insurance & Trust Co. (chartered in 1830) and continued under title of Bank of New York & Trust Co. until July 1, 1938 when title of Bank of New York was resumed. Effective May 1, 1948, merged Fifth Avenue Bank of New York (incorporated in 1875) and changed name to Bank of New York & Fifth Avenue Bank. Consideration was issuance of 20,000 $100 par shares on basis of 4 shares for each $100 share of merged bank. On Apr. 29, 1952, resumed name of The Bank of New York.

On Dec. 7, 1966 merged Empire Trust Co. (incorporated in 1901) by exchange of 296,150 shares of common and 296,150 shares of new preferred on basis of 1 common and 2 new convertible preferred shares for each of 148,075 outstanding Empire common shares.

Conducts a general banking and trust business. Member of Federal Reserve System and Federal Deposit Insurance Corp., Bank and trust services are provided to customers through 143 offices in New York and three overseas offices in London, Singapore and Grand Cayman. The Fifth Avenue Bank Safe Deposit Vaults, Inc., a wholly-owned subsidiary, was merged Jan. 31, 1967.

On Jan. 1, 1976 the eight member banks of The Bank of New York Co., Inc. were merged into the Bank of New York.

On Sept. 11, 1980, Co. acquired and merged Empire National Bank. Seven new subsidiaries were also acquired in connection with the merger along with The Bank of New York Company, Inc. acquiring two mortgage banking subsidiaries. Cost of these transactions was $28,081,000.

Subsidiaries: BNY International Investments, Inc., an Edge Act corporation, was formed Apr. 13, 1966; Fourten South Broadway Corp.; Valley Premises, Inc.; Robank, Corp.; Empro Properties, Inc.; Meadows Land Co.; Karpen Properties, Inc.; Gram Brent Corp.; Ca-Tar Properties; Orechester Inc.; Empark, Inc.; The Bank of New York International, Inc., and Edge Act corporation, was formed Dec. 2, 1980.

Senior Management

J.C. Bacot, Chmn. & Chief Exec. Off.
P. Herrick, President & Chief Oper. Off.
D.D. Papageorge, Sr. Exec. Vice-Pres. & Chief Fin. Off.
G.W. Spurr, Jr., Sr. Exec. Vice-Pres. & Chief Credit and Comm'l Banking Off.
S. Harrison, Sr. Exec. Vice-Pres. & Chief Oper. Off.
D.F. Adams, Exec. Vice-Pres. & Chief Trust Off.
N.P.S. Merrill, Exec. Vice-Pres.—U.S. Commercial Banking
R.D. Field, Exec. Vice-Pres.—Reg. Banking
C.L. Booth, Exec. Vice-President & Chief Invest. Off.
R.E. Keilman, Senior Vice-Pres. & Comptr.
R.J. Goebert, Vice-Pres. & Sec.

Senior Vice Presidents

W.A. Anderson, Jr.
G.W. Bennett
W.H. Cushman
H.V. Delapena, Jr.
G.H. Denniston, Jr.
M.L. Dinerstein
Emmett Driscoll
J.F. Fadell
R.R. Gardella
A.R. Griffith
J.C. Hoopes
J.P. Isoldi
P.C. Karda
W.J. Kemple
R.C. Kuser, Jr.
J.B. Mencke
J.E. Otis, Jr.
O.M. Quattlebaum
T.A. Reny
L.M. Richey, Jr.
J.A. Ross
F.B. St. John, Jr.
R.P. Ulin

Directors

H.A. Ashforth
J.C. Bacot
J.G. Brookhuis
Alexander Calder, Jr.
E.W. Debevoise
H.A. Eckmann
R.E. Gomory
P. Herrick
Lawrence Lachman
Robert Luciano
A.I. Mendolia
D.L. Miller
H.B. Morley
E.J. Mortola
J.G. Phillips
H.L. Plumley
R.T. Quittmeyer
Catherine A. Rein
D.C. Staley
J.R. Stevenson
E.J. Sullivan
W.L. Wearly

No. of Employees: Dec. 31, 1984, 7,454.

Main Office: 48 Wall St., New York, NY 10015. **Tel.:** (212)530-1784.

COMPARATIVE STATEMENT OF CONDITION, AS OF:

(in thousands of dollars)

	12-31-84	12-31-83	12-31-82	12-31-81	12-31-80	12-31-79	12-31-78
Assets:							
Cash and due from banks	866,552	1,030,819	870,465	904,714	980,709	1,076,707	1,128,274
Interest bearing deposits in banks	1,016,378	989,643	1,697,928	1,438,201	1,814,162	1,710,900	1,101,646
U.S. Government obligations	521,620	494,830	326,571	337,982	307,936	180,984	231,629
U.S. Govt. Agency obligations	53,850	53,135	63,405	78,291	89,793	97,642	96,913
Oblig. of States & Political subdivs.	783,281	647,049	586,560	606,687	632,773	556,497	576,965
Other securities	26,473	29,920	36,938	37,635	33,600	22,795	23,710
Trading account securities	66,595	32,263	50,058	3,857	2,168	16,967	38,346
[1]Loans	9,090,442	7,625,363	7,392,972	6,905,272	5,658,664	4,295,001	3,503,886
Lease financings			50,952	46,477	34,312	19,000	6,765
Federal funds sold & secur. pur. und. resale agree.	419,415	261,400	326,825	50,900	112,334	459,485	306,458
Fed. fds. sold to affil. bk.	788,600	379,841	......	......	......	......	......
Accrued interest receivable	194,939	151,313	158,054	175,982	165,495	121,470	78,145
Other assets	113,335	132,633	174,368	189,194	219,537	99,348	38,531
Customers' accept. liability	659,646	700,576	772,983	508,626	213,344	265,949	100,294
Premises & equipment	115,353	120,421	117,556	127,108	107,401	67,921	46,446
Total	14,716,429	12,649,306	12,625,635	11,410,866	10,372,228	8,990,666	7,278,008
Liabilities:							
Deposits:							
Domestic offices:							
Non-interest bearing	2,869,873	2,849,191	2,660,436	5,631,840	5,691,934	4,703,426	4,373,184
Interest bearing	4,047,105	4,036,106	2,908,161				
Foreign offices:							
Non-interest bearing	3,921	4,029	2,685	2,448,425	2,627,231	2,166,911	1,553,229
Interest bearing	3,579,990	2,675,508	3,502,855				
Total deposits	10,500,889	9,564,835	8,874,136	8,080,265	8,319,165	6,870,337	5,926,413
Accrued taxes & other expenses	84,708	108,460	67,794	41,863	27,940	27,794	24,708
Accrued interest payable	122,076	108,792	116,902	136,744	131,445	89,972	52,644
Federal funds purch., secur. sold under repurch. agreemts. & other sh.-tm. borrow.	2,605,370	1,476,153	2,190,807	2,089,416	1,202,550	1,310,117	777,310
Other liabilities	60,776	53,447	49,139	39,910	48,361	10,871	8,940
Acceptances outstg.	659,646	700,738	772,989	509,201	213,587	265,949	100,295
Common stock	112,272	112,272	112,272	112,272	112,272	62,373	62,373
Surplus	158,531	158,531	123,531	123,531	73,531	123,430	123,430
Undivided profits	412,161	366,078	318,065	277,664	243,377	229,823	201,895
Total capital accounts	682,964	636,881	553,868	513,467	429,180	415,626	387,698
Total	14,716,429	12,649,306	12,625,635	11,410,866	10,372,228	8,990,666	7,278,008

[1]Shown net of unearned income of $176,760,000 at Dec. 31, 1984; $120,470,437 at Dec. 31, 1983; $102,222,483 at Dec. 31, 1982; and net of allowance for loan losses of $129,270,000 at Dec. 31, 1984; $82,000,000 at Dec. 31, 1983; $64,100,000 at Dec. 31, 1982; $44,300,000 at Dec. 31, 1981; $36,800,000 at Dec. 31, 1980; $26,100,000 at Dec. 31, 1979; $25,896,327 at Dec. 31, 1978; $25,503,300 at Dec. 31, 1977.

Capital Stock: Authorized and outstanding, Dec. 31, 1984, $112,271,940; par $27.

All owned by The Bank of New York Co., Inc.

Par changed from $100 to $40 July 24, 1963 by 2½-for-1 split; from $40 to $20 par June 29, 1966 by 2-for-1 split and from $20 to $15 par Dec. 7, 1966 share-for-share on reclassification; from $15 to $27 par Oct. 14, 1980.

Dividends (payments since 1921 follow):

On $100 par shares:

1922	$10.00	1923	$20.00	1924-25	$21.00
1926	24.00	[1]1927	22.00	1928	18.00
1929-31	20.00	1932	16.00	1933-49	14.00
1950-52	16.00	1953-54	18.00	[1]1955	12.50
1956	12.00	[1]1957-58	14.00	1959	12.00
1960-62	13.00	1963	9.75		

On $40 par shares:

[1]1963	1.20	1964	5.25	1965	5.40
1966	4.20				

On $20 par shares:

1966	0.70	1967	0.58

On $15 par shares:

1967	2.22	1968	1.00	[2]1969	2.02

[1]Also paid stock dividends 1927, 50%; 1955, 100% ($4 cash paid prior to stock dividend); Oct. 1, 1958, 50% (after cash dividends); July 24, 1963, 11.11%.

[2]To June 17; last publicly paid dividend.

Capital Changes: Capital stock was increased from $4,000,000 to $6,000,000 Apr. 12, 1927, by a 50% stock dividend May 3, 1927; to $8,000,000 upon merger of Fifth Avenue Bank, effective May 1, 1948; to $16,000,000 by 100% stock dividend Mar. 25, 1955; $24,000,000 by 50% stock dividend Oct. 1, 1958; to $27,000,000 by sale of stock Nov. 19, 1958; to $30,000,000 July 24, 1963 by 11.11% stock dividend; to $49,153,500 Dec. 7, 1966 on merger of Empire Trust Co. (for details see above). Decreased to $49,147,500 Oct. 15, 1968 on conversion of 100 preferred shares and to $31,384,500 May 29, 1969 on conversion of 296,050 preferred shares. Increased to $62,373,300 on Jan. 1, 1976 due to merger. Increased to $112,271,940 on Oct. 14, 1980 due to an increase in par value, transferred from surplus.

THE BANK OF NEW YORK COMPANY, INC.

200 YEARS OF LOOKING AHEAD
THE BANK OF NEW YORK
1784-1984

1984 Annual Report

Financial Highlights

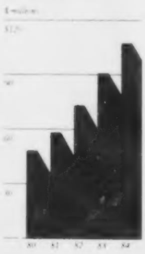
Dollars in thousands except per share amounts

	1984	1983
For the Year		
Net Income	$107,520	$90,612
Cash Dividends on Common Shares	28,978	25,468
Return on Average Assets	.80%	.71%
Return on Average Common Equity	15.71	15.43
Allowance for Loan Losses as a Percent of Loans	1.27	1.01
Primary Capital as a Percent of Assets	6.34	5.83
Per Common Share		
Primary Earnings	$ 6.40	$ 5.70
Fully Diluted Earnings	5.36	4.72
Cash Dividends	1.89	1.73½
Book Value at Year End	41.78	38.04
At Year End		
Assets	$15,156,758	$12,797,199
Loans	10,303,329	8,140,487
Investment Securities	1,480,913	1,281,761
Deposits	10,527,969	9,559,648
Common Shareholders' Equity	660,262	564,003
Total Shareholders' Equity	760,262	664,003
Common Shares Outstanding	15,786,529	14,807,438
Common Shareholders of Record	13,648	12,810
Employees	7,457	8,039

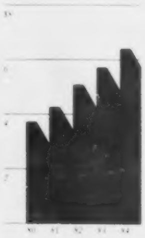
Contents

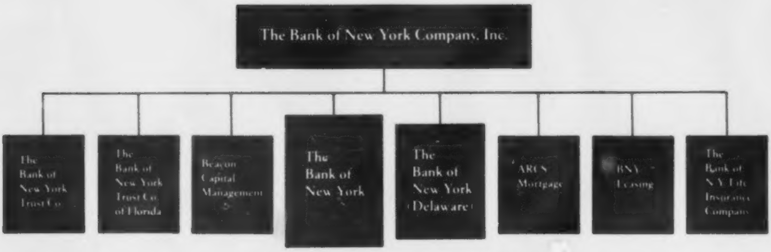
Financial Highlights 1
The Bank of New York Company's Principal Subsidiaries 2
Letter to Our Shareholders 3
Commercial Banking 8
Regional Banking 12
Trust and Investment 16
Securities Processing 18
ARCS Mortgage 20
Beacon Capital Management 20
Bicentennial Review 21
Financial Section 23

Net Income Before Extraordinary Charge



Primary Earnings Per Common Share




The Bank of New York Company, Inc.
The Bank of New York Trust Co.
The Bank of New York Trust Co. of Florida
Beacon Capital Management
The Bank of New York
The Bank of New York (Delaware)
ARCS Mortgage
BNY Leasing
The Bank of N.Y. Life Insurance Company

The Bank of New York Company's Principal Subsidiaries

The Bank of New York, established in 1784, is a money-center bank of worldwide stature with assets in excess of $14 billion.

The Bank of New York (Delaware) began operations in December 1983. At year end 1984, its assets exceeded $1 billion.

The Bank of New York Trust Company, incorporated in 1981, develops, markets and services master trust accounts.

The Bank of New York Trust Company of Florida, N.A., headquartered in Miami, principally serves Florida residents. It also has authority to provide trust services in other states.

Beacon Capital Management Company, Inc., established in 1983, manages the assets of large tax-exempt institutional accounts, principally corporate pension funds.

ARCS Mortgage, Inc. services a mortgage portfolio of $1.6 billion. Its rank in size among U.S. mortgage banking companies rose to 74th in 1984 from 93rd in 1983.

BNY Leasing, Inc. provides equipment lease financing to major corporations.

The Bank of New York Life Insurance Company, Inc. reinsures credit life and accident and health insurance sold in connection with consumer loans.

To Our Shareholders:

It is a pleasure to report that 1984 was the seventh consecutive year of record earnings for our Company. All major business areas contributed to this result.

Net income of $107.5 million was 18.7% higher than in 1983. Earnings per share, at $6.40, increased 12.3% from the 1983 level. Dividends were raised in October by 10.9% to an annual rate of $2.04. This was the eighth consecutive annual dividend increase.

Dividends Paid Per Common Share

The compound growth rates of net income and earnings per share over the past five years were 19.8% and 12.6%, respectively. Over this period, the growth rates of net interest income and noninterest income alike exceeded those of most money-center banking companies. A major goal was to strengthen our profitability and balance sheet ratios. In 1984 these ratios were among the best in the money-center group. Rates of return on assets and common shareholders' equity were 0.80% and 15.71%, respectively. The ratio of primary capital to total assets, an important measure of financial strength, was 6.34% at year end. After the end of the year, we strengthened our primary capital further by calling for redemption the Company's 6¼% and 12% convertible debentures as of February 15, 1985. If all of the debentures are converted into common stock, rather than redeemed, shareholders' equity would increase by $85.4 million. Had this amount of additional equity been outstanding at the end of 1984, our primary capital ratio at that time would have been 6.90%.

Return on Average Common Equity

Commercial loans rose 27.8%, both as a result of strength in the economy and as a result of our continuing ability to penetrate selected segments of the commercial banking market. A rising loan limit has enabled The Bank of New York to be a prime lender to a growing number of large corporations, and we are placing new emphasis on loans to medium-size corporations in the greater New York metropolitan area. Comparatively favorable funding costs have made it possible for us to price loans very competitively, and the ability to offer dependable cash management services to corporations has been helpful in gaining new relationships. Our specialized lending divisions continued to contribute to the growth of the loan portfolio. Especially strong were the contributions of divisions specializing in loans to the securities industry; communications, entertainment and publishing companies; and public utilities.

In order to accommodate the growth of our business, we took the following steps in 1984:

- We created a Capital Markets Division to develop and market sophisticated corporate finance, leasing and treasury-related services to domestic and multinational customers.

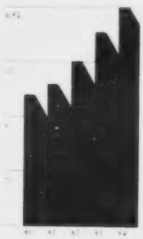
- We enlarged further our cash management capabilities to stay at the forefront of this important area of service to corporations.
- We replaced the former Metropolitan and National divisions with a U.S. Commercial Banking Sector consisting of six divisions. Four of these divisions have geographic responsibilities: New York Corporate, Eastern, Northeastern and Western. The other two are our Securities Industry and Public Utilities lending groups, which were made divisions in recognition of their importance and prospects.
- The Bank of New York (Delaware), which serves corporations in the Mid-Atlantic states, opened a Cayman Islands branch in the summer, giving it greater flexibility in servicing the needs of its customers. In the fall, it formed its own Construction and Real Estate Lending group.
- In September and December, respectively, we opened loan production offices in Los Angeles and Stamford, Connecticut, to help us serve more effectively than before corporate customers in California and New England.
- We obtained approval from the Comptroller of the Currency to open consumer banks in Florida and New Jersey.
- Expansionary steps taken during the year in other business areas included the transfer of our credit card operations to a new state-of-the-art facility in Delaware; the opening of a trust representative office in Naples, Florida, by our Miami trust subsidiary; and the opening of three new offices by our ARCS Mortgage subsidiary.

Overseas, we continued to emphasize loans to subsidiaries of multinational corporations in industrially advanced countries, including principally the United Kingdom, continental Europe and Australia. The only developing countries to which we have major exposures are Mexico and Brazil. These countries made considerable progress during 1984 in improving their balance of payments positions with assistance from banks, governments and multinational agencies. We feel that their financial conditions will continue to improve.

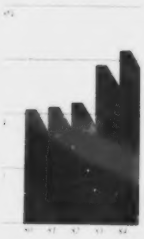
Retail banking also enjoyed an excellent year. Consumer loans rose substantially, led by a particularly strong increase in automobile loans. We reduced interest rates on auto loans to the lowest level available from major banks in New York State. As a result, the volume of these loans rose steadily throughout the year instead of showing the seasonal pattern experienced in prior years.

We completed in the first half of 1984 the repositioning of our retail banking network through the sale of 36 branches in the Albany, Buffalo, Endicott and Olean regions that were marginally profitable for us, at a $24 million capital gain. In upstate New York, we retain six branches in the Syracuse area, from which we continue to serve individuals and middle-market customers. This repositioning now allows us to

Return on Average Assets



Primary Capital Ratio



concentrate our retail banking resources in the greater New York metropolitan area, where most of our branches are located, our market position is strong, and growth potential is good. The capital gain from the branches we sold was used primarily to increase the loan loss reserve.

Trust fees rose 5.4% from the 1983 level to $152.9 million. Securities processing and investment management activities both contributed to that increase. In the securities processing area, substantial gains came from government securities clearance, institutional custody and corporate agency activities. In the Trust and Investment sectors, the strongest gains came from personal trust and investment advisory activities.

In the past two years, a number of states have formed regional compacts permitting interstate banking, but excluding banks from New York and other states that are not members of those compacts. An example is Connecticut, which passed a law allowing entry by banks from certain other New England states only. That law does not permit consummation of our proposed acquisition of Northeast Bancorp.

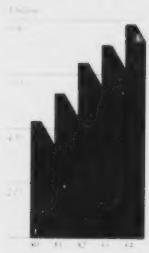
We are encouraged that the U.S. Supreme Court has agreed to decide whether regional banking compacts are constitutional, and we hope that Congress will address the issue of interstate banking in a manner that does not discriminate against New York banks. Confident that we can compete effectively in a fully deregulated environment, we favor removal of remaining geographic and product constraints that limit the areas and activities in which bank holding companies can engage.

The Company's strong growth over the past several years and results in 1984 reflect the cooperation of all our employees. We base our confidence in the future on their professionalism and dedicate to them this annual report for the year of The Bank of New York's Bicentennial.

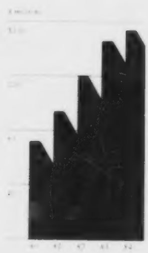
We are very proud of the public recognition we received in connection with our Bicentennial. Events we sponsored to mark this historic milestone are reviewed on pages 21 and 22 of this report, together with commendations from the mayor of New York City and the governor of New York State and tributes read into The Congressional Record. Many shareholders and other friends of The Bank of New York have sent us their congratulations. We take this opportunity to thank them.

On February 13, Edward L. Hennessy, Jr., chairman and chief executive officer of Allied Corporation, will be joining our Board of Directors. His very successful business background and highly regarded management perspectives will make him an outstanding addition to our Board.

Average Loans



Trust Fees



We want to express our appreciation for the valuable guidance and support we receive from our Board of Directors. We also want to thank our customers and shareholders as well as our employees for their loyalty and contribution to our progress.

A final, special note:

Messrs. H. Adams Ashforth, Eli Whitney Debevoise and H. Ladd Plumley have announced their intention to retire from our Board of Directors, on which they have served for many years. Mr. Debevoise joined the Board of The Bank of New York in 1938. Messrs. Ashforth and Plumley joined in 1944 and 1957, respectively. Mr. Debevoise was elected to the Board of the holding company at its inception in 1968, and Messrs. Ashforth and Plumley were elected to that Board in 1976.

Since they became directors, vast changes have taken place in conditions affecting the economy, commerce and banking. Throughout those years, their valuable counsel has contributed to our Company's expansion and prosperity. We want to express to them our deep appreciation for the important contributions they have made to our progress.

J Carter Bacot

J. Carter Bacot,
Chairman and Chief Executive Officer

Peter Herrick,
President

February 1, 1985

Commercial Banking

Our portfolio of commercial loans increased 27.8%. All divisions in the Commercial Banking Sector contributed to this performance. The Sector's growth continued to reflect the primary strategy of increasing our share of lending to the nation's 1,500 largest corporations and expanding business within specialized lending areas, while maintaining strict credit-quality standards.

Account officers of the U.S. Commercial Banking Sector focus on the credit requirements of large corporate customers in various regions of the country.

Our ability to price loans very competitively and to offer high-quality cash management services was helpful in establishing new relationships. Also, as in the past, the expertise of our specialized industry lending divisions contributed importantly to the growth of our loan portfolios and commercial fee income:

- The Securities Industry Banking Division achieved considerable growth by increasing the volume of business with existing borrowers and adding to its roster of customers, which now number about 100 firms, including 22 of the 25 largest U.S. securities broker/dealers.
- The Mortgage and Construction Lending Division also enjoyed another record year. Average loans and fees were at all-time highs, and we made a substantial amount of high-quality loans in diverse locations. In addition, a Construction Lending Group was established by our new Delaware bank to enhance our position in the middle Atlantic states.
- The Oil, Gas and Extractive Industry Division retained a good portfolio, despite a decline in the overall quality of energy-related lending opportunities. The Division nonetheless gained several large new customers, primarily as a result of acquisition financings.
- Through the Communications, Entertainment and Publishing Division, we continued to be a principal provider of credit and financial services to these rapidly growing industries. This has been one of our best growth areas over the past few years, and the Division's loans were at record levels in 1984. Cable television companies, broadcasters and publishing companies all required substantial funds to expand their businesses and finance acquisitions.
- The Public Utilities Division, the newest of our special industry units, showed the biggest gain. Its assets grew more than 50% over the 1983 level. It now has a diversified portfolio of loans and commitments to investor-owned public utilities and electric cooperatives operating in 26 states, compared to 20 states in 1983.

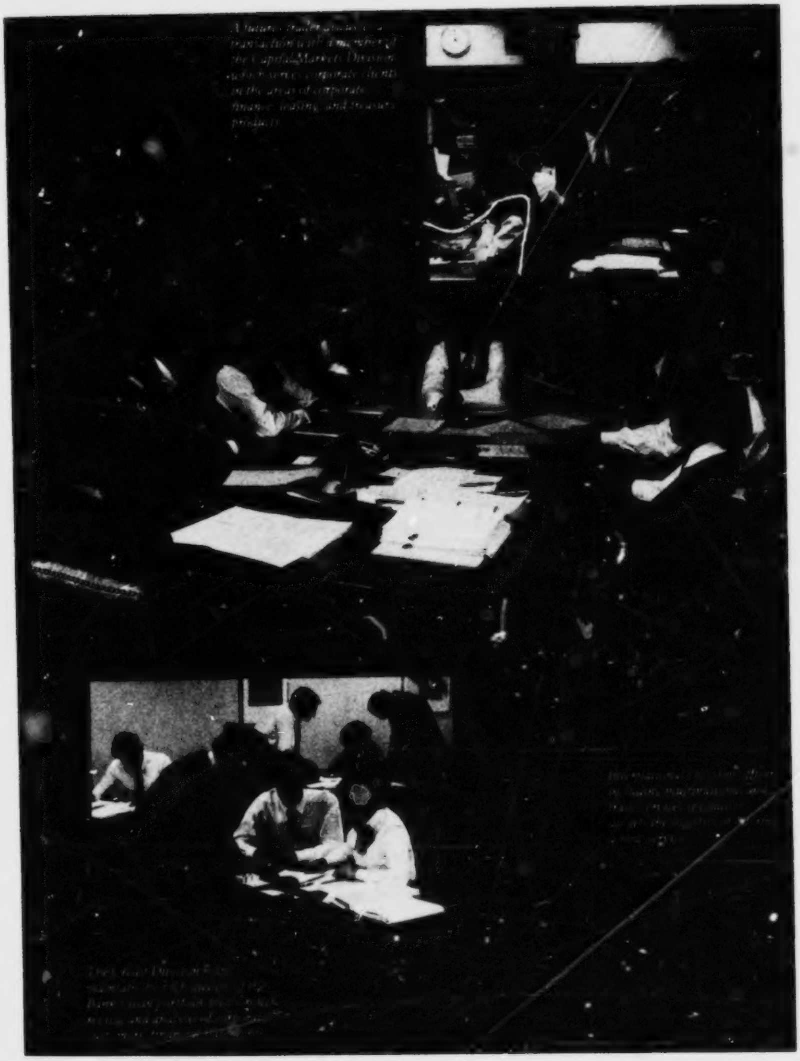
Cash management sales representatives and marketing staff discuss new brochures which describe services that help corporations maximize use of their cash flow.

Officers of the Communications, Entertainment and Publishing Division serve the credit and non-credit needs of cable companies, broadcasters and publishing companies.

Members of the Middle Market Department serve the banking needs of medium-size companies in the greater metropolitan New York area.

The International Division continues to concentrate on expanding its business in the United Kingdom, continental Europe and the stronger developed and developing countries in the Far East, including particularly Australia through our two merchant banking affiliates in that country. As in the past, special focus is concentrated worldwide on multinational corporations and their local affiliates. We continue to monitor Latin American exposures closely, and we are encouraged by the financial stabilization programs already in place or in advanced stages of negotiation.

During 1984, we strengthened and expanded our commercial banking organization to prepare us for the increase in business we see in the years ahead. We replaced the former Metropolitan and National divisions with a U.S. Commercial Banking Sector, which includes six divisions that are focused on the particular needs of specific regions or industries in the United States. To support and supplement the expanded Commercial Banking Sector, we further enlarged our cash management operations and created a new Capital Markets Division to develop and offer a wide range of sophisticated corporate finance, leasing and treasury-related services. In addition, we have centered in our new Delaware bank our corporate banking activities in the Mid-Atlantic area. Also, we opened loan production offices in Los Angeles and Stamford, Connecticut, to enhance our servicing and prospecting capabilities in California and New England, where growth prospects are especially attractive.



A futures trader [illegible] a transaction with a member of the Capital Markets Division which serves corporate clients in the areas of corporate finance, leasing, and treasury products.

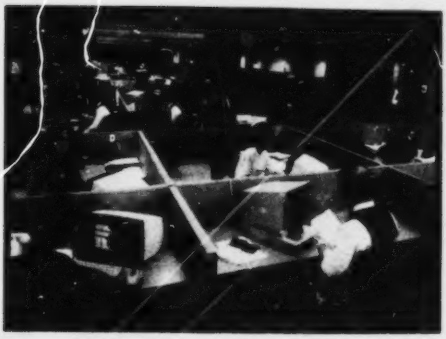
[illegible]

Trust [illegible] Division [illegible] Bank's [illegible] trading and [illegible]

Regional Banking

Progress was made during 1984 in improving further the efficiency of our branch system.

Consumer loans, commercial loans and fees generated by our branches all rose substantially from 1983 levels, reflecting:

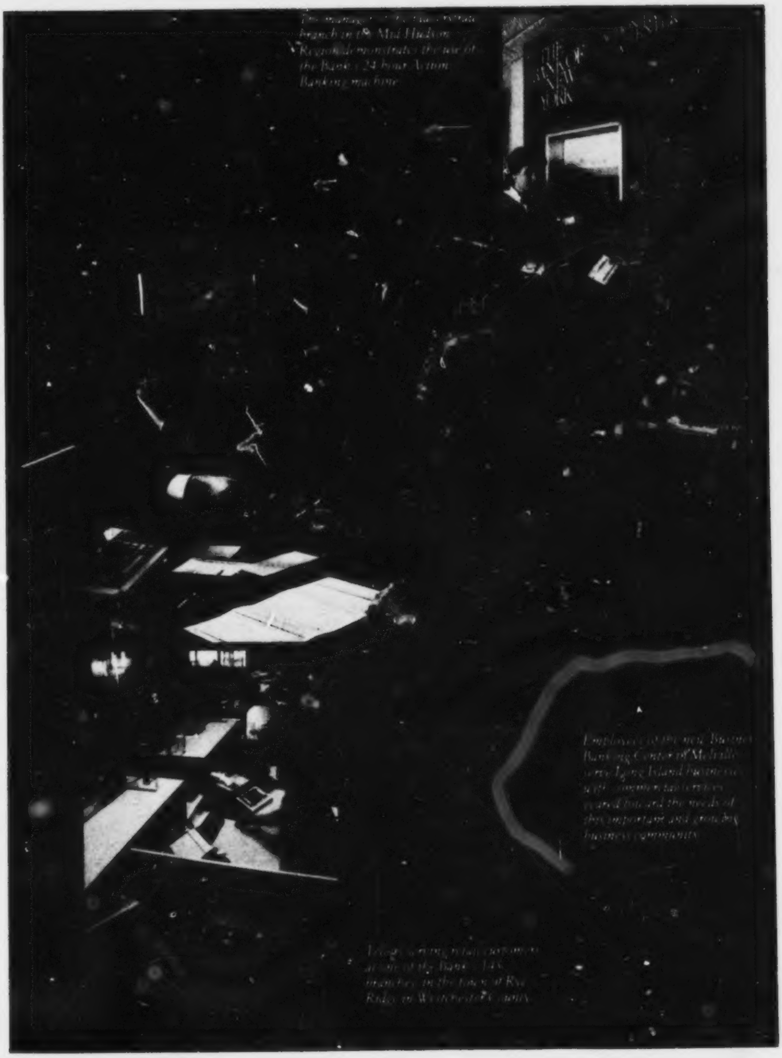


The automated credit approval system in Delaware helps us respond to credit card applications quickly.

- Good economic growth in most of the communities we serve.
- Interest rate reductions on consumer loans, led by reductions on automobile loans to the lowest level available from major New York banks. Average consumer loans on the books of our present branches were up 33.4%, and in the last quarter of the year, these branches booked 2.6 times as many auto loans as in the year-earlier period.
- Increased emphasis on lending to medium-size and smaller corporations, combined with competitive loan pricing and success in selling ancillary cash management and international trade services. CheckInvest, which links the checking accounts of small businesses with money market funds, remains the only service of its kind in our marketplace and a highly effective vehicle for developing new business relationships.

We continued to expand our electronic banking program designed to improve quality of service and reduce costs:

- We began implementation of a system giving tellers and platform officers instantaneous access to account information on visual display screens. At year end, 84 of our 143 branches were online, and most of the remainder are scheduled to be online by mid-1985.
- We purchased 14.4% of a shared automated teller machine (ATM) network being set up in conjunction with seven other banks. The network, expected to become operational within a few weeks, will make the use of over 800 ATMs available to our customers, initially in the states of New York and Connecticut.



The manager of the [illegible] branch in the Mid-Hudson Region demonstrates the use of the Bank's 24-hour Action Banking machine.

Employees in the new Business Banking Center of Melville serve Long Island businesses with commercial services geared toward the needs of this important and growing business community.

Tellers serving retail customers at one of the Bank's 14[illegible] branches in the town of Rye Ridge in Westchester County.

Custom Bankers provide highly personalized banking services to affluent individuals.

In addition, we took several steps to improve the organizational structure of our branch network:

- The sale of 36 branches in areas of upstate New York where our market penetration was low now enables us to focus more of our marketing resources on the greater New York metropolitan area, where most of our branches are located. We continue to serve customers in upstate New York through six branches in and near Syracuse.
- Our Credit Card Division was moved to a state-of-the-art facility in Delaware. This gives us additional capacity to process credit cards at a comparatively favorable cost.
- We centralized our Consumer Loan Administration, thereby eliminating the redundancy of separate units performing this function in each region.
- We created a Commercial Loan Administration Division in White Plains to upgrade the quality and control of regional commercial lending activity.
- We obtained approval from the Comptroller of the Currency to open consumer banks in Florida and New Jersey.
- We enhanced our Custom Banking Service, tailored to specific needs of affluent individuals. We expanded our Custom Banking staff and created a department to coordinate its activities on a bankwide basis.
- We initiated a program of expanding our presence on Long Island. After year end, we opened a Business Banking Center in Melville, dedicated to serving the needs of middle-market corporate borrowers. Also, we acquired a sizable full-service branch in Great Neck from Hamburg Savings Bank. This branch, which will open in April, will extend our branch marketing activities into an area that has excellent potential for future growth.

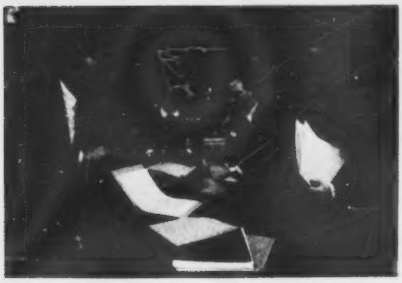
Action Phone makes getting a loan or account information fast and convenient.

Sophisticated check-processing equipment increases operating efficiency by simultaneously photographing and encoding transactions.

Trust and Investment

Fees from trust and investment services were at record levels, and we continued to enhance the scope and quality of these services.

Fee income in our Trust Sector increased as a result of new business, greater market activity and selective fee adjustments. In addition, we enhanced the level of service to our customers through systems and organizational improvements. Not reflected in 1984 results, but certain to have a very positive effect beginning in 1985, was the passage of legislation in New York State that frees corporate trustees from having their fees fixed by the legislature. This will, we feel, help us expand further and improve the profitability of our trust activities.



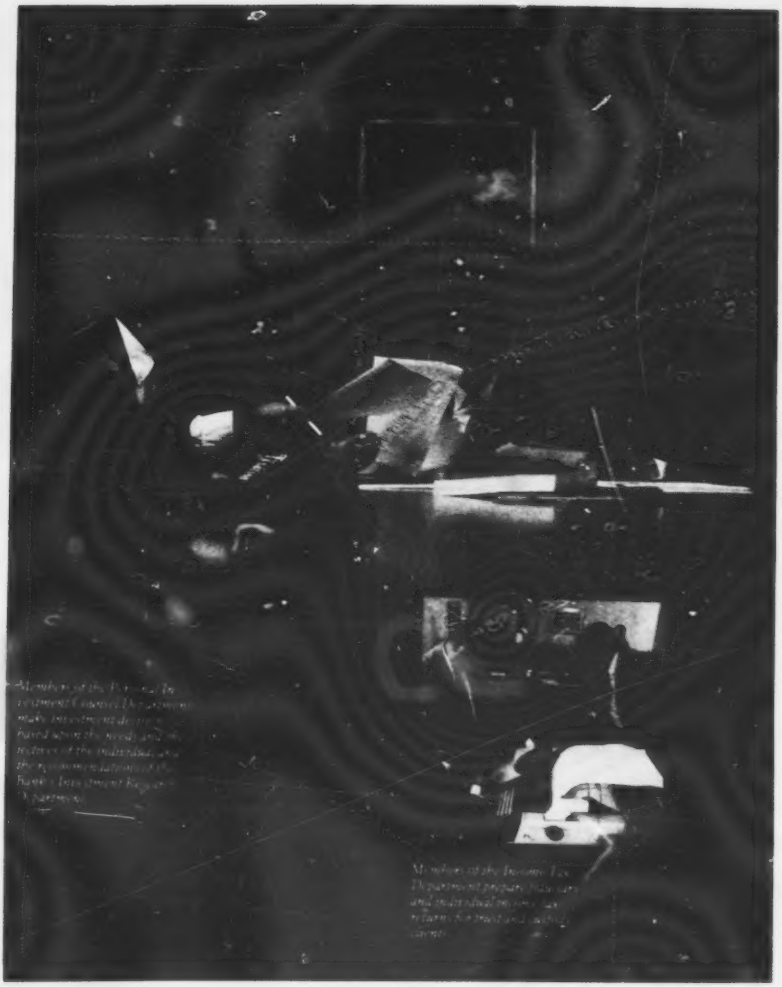
Members of the Trust Division discuss the provision of trust and investment services to non-U.S. residents through the International Personal Trust Service.

Outside New York City, steps were taken to expand our trust business. We added a trust development officer in the Westbury office on Long Island; and our trust company in Miami, created in 1983, opened a trust representative office in Naples, Florida. Initial response to these steps has been favorable.

Increases in revenues and profitability were achieved by the services of our Personal Investment Division: personal investment counsel, tax-exempt bond management and financial planning. We began to offer personal investment counsel in our White Plains office during the year. Financial planning enjoyed particularly strong recognition from corporations on behalf of their executives.

We completed the restructuring of our Equities Research Department, which has enabled us to concentrate our resources exclusively on obtaining the best possible investment results for the portfolios we manage in-house. We believe this restructuring contributed to our successful performance in 1984.

The Bank's institutional investment services continued to enjoy good acceptance. Revenues from fixed-income investment management were up sharply from the 1983 level, and further gains are expected in 1985. We added two new strategies to our broad option-writing capability. These techniques have been well received in the marketplace and we expect significant new business to develop from this source.



Members of the Personal Investment Counsel Department make investment decisions based upon the needs and objectives of the individual, and the recommendations of the bank's Investment Research Department.

Members of the Income Tax Department prepare fiduciary and individual income tax returns for trust and custody clients.

Securities Processing

Securities processing is a major business in which The Bank of New York has achieved a position of prominence. Today, we are a leader in the servicing of mutual funds, we are the largest processor of unit investment trusts, and we lead all competitors in government securities clearance. We are also committed to maintaining strong positions in institutional custody, master trust and various corporate agency services such as trustee, tender agent, registrar, transfer agent and dividend disbursement agent.

Fees from securities processing have contributed importantly to our growth over the past several years and account for a larger portion of our operating income than is the case for most other banks. This gives our Company an important degree of earnings diversification. Prominence in securities processing is also helpful in gaining new lending and investment management relationships.

Income from securities processing activities was at a record level in 1984. The largest increase was achieved in clearing securities for brokers and dealers, primarily government instruments. Unit investment trust services, corporate agency activities, institutional custody, master trust and securities lending contributed to the increase, as did two new services introduced during the year: the processing of Treasury Receipts and American Depository Receipts for securities of foreign corporations.

Employees of the Mutual Funds Division receive telephone instructions from shareholders regarding purchases and redemptions in, and exchanges between, their fund accounts.

During the year, we successfully installed various phases of two important state-of-the-art systems that will enable us to improve the quality of our product and handle a large increase in volume at relatively moderate cost:

- A new automatic call distribution system makes it possible for us to answer telephone inquiries from customers of our mutual funds and unit investment trust departments more promptly and efficiently than we could with our previous system.
- We installed the first phase of an automated securities processing system designed to speed and more accurately control the movement of securities between the Bank and other institutions. Receipt and delivery of securities are already conducted through the new system. Other activities such as income collection are scheduled to be added to the system in 1985.

These, as well as other computer-based systems, are designed to permit modifications and additional functions as the requirements of the markets we serve evolve. We have the managerial experience, technical skills, scale of operations and attendant cost efficiencies to stay at the forefront of the expanding securities processing business.

ARCS Mortgage

Despite volatile mortgage rates, our ARCS Mortgage subsidiary continued its expansion in 1984. At year end, it serviced $1.6 billion of residential mortgages, an increase of approximately 28% over the year-earlier level. Its rank in size among U.S. mortgage banking companies rose to 74th from 93rd.

During the year, ARCS Mortgage opened new offices in Seattle, Washington, Tustin, California, and Chicago. It introduced new mortgage instruments and created a division to specialize in the financing of multifamily residential buildings. The new Chicago office specializes in the placement of commercial mortgages. After year end, ARCS opened two Home Loan Centers on Long Island, bringing to three the number of its offices in that area.

ARCS Mortgage representatives in a Long Island office discuss the financing of a local condominium project.

Beacon Capital Management

Beacon Capital Management made significant progress last year. This new company was established in 1983 as a wholly owned, but independently operated, subsidiary to manage the investment portfolios of select, large tax-exempt accounts, including corporate pension funds and endowment funds.

Faced with volatile and trendless financial markets in 1984, Beacon raised substantial cash reserves early in the year in the equity accounts it manages. As a result, the investment performance of the equity accounts it manages was superior to that achieved by most competing fund managers. In balanced accounts, Beacon increased the portion of funds allocated to bonds. The success of these strategies has been and will continue to be helpful in Beacon's new business activities.

Initially housed in The Bank of New York's building at 48 Wall Street, Beacon will move during the first quarter of 1985 into offices of its own in New York's World Trade Center.

Beacon Capital Management portfolio managers invest the assets of tax-exempt institutional portfolios such as corporate retirement, profit-sharing and thrift plans.

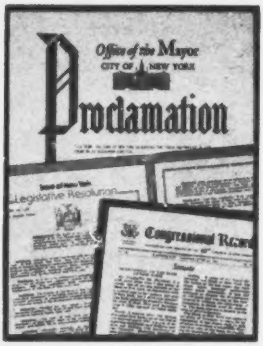
Bicentennial Review

Chairman J. Carter Bacot and Mayor Edward Koch of New York City, accompanied by President Peter Herrick and Lieutenant Governor Alfred B. Del Bello, display the Mayor's "Bicentennial Day" Proclamation.

The celebration of our Bicentennial in 1984 marked a truly historic milestone for the Bank. We chose as our theme, "200 Years of Looking Ahead," which reflects a proud heritage that has prospered as a result of generations of investors and managers with a clear and constant vision towards the future. We were proud of and touched by the many congratulatory wishes we received from shareholders, customers and the general public. And we are most appreciative of the recognition accorded the Bank by national, state and local public officials throughout the year.

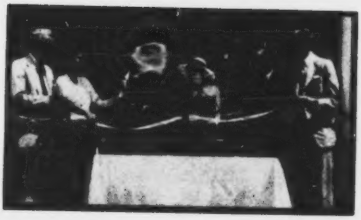
Mayor Koch of New York City proclaimed June 9th, the actual date of our founding in 1784, as "The Bank of New York Bicentennial Day." Also on that day, the Bank was honored to participate with the National Park Service in a dedication ceremony for the Hamilton Grange National Memorial. The Grange, located at 141st Street and Convent Avenue in New York City, was the "country" home of Alexander Hamilton, the Bank's founder.

The Bank was recognized in Congress by the reading of tributes into The Congressional Record by Senators Alfonse D'Amato and Daniel Patrick Moynihan. In recognition of our contribution to New York State, Governor Mario Cuomo issued a citation and State Senator John Flynn introduced a resolution in the New York State Legislature.



A proclamation in which "The Bank of New York Bicentennial Day" is declared by Mayor Edward Koch in New York City; a Legislative Resolution signed by Governor Mario Cuomo of New York State honoring the Bank's anniversary; and The Congressional Record documenting the event of the Bank's Bicentennial.

We were pleased to underwrite an historical exhibit entitled "The Bank of New York and Early New York Banking." This 90-piece exhibit features many documents and artifacts from the Bank's own archives, as well as memorabilia from museums and other financial institutions. The exhibit was displayed at the New York Public Library, the Museum of the City of New York, Adelphi University, White Plains, the



The Bank participated in the official reopening and dedication of the former home of Alexander Hamilton, now known as the Hamilton Grange National Memorial.

U.S. Military Academy at West Point and Hamilton College.

The Bank's rich history was vividly captured in a corporate biography entitled *200 Years of Looking Ahead.* Written by Herbert Parmet, a distinguished professor of history, the book chronicles the Bank's ability not only to survive but to prosper through many economic crises and adapt to a constantly changing financial marketplace. We made the book available to shareholders and employees and presented it as a gift to customers and public officials who attended the various receptions that were held throughout the year to commemorate our Bicentennial.

Looking back over the year we are very proud of the recognition that our Bicentennial has received and sincerely thank all who shared in its celebration by their expression of good wishes. As we enter our third century of operation, we remain committed to the tradition of quality service and profitable growth that has been upheld throughout our 200-year history.

The Bank's Bicentennial exhibit, which traces the history of early New York banking, was displayed throughout the state at libraries, museums, and other public locations.

The Bank hosted a Bicentennial reception for foreign banking and finance officials prior to the International Monetary Fund meetings.

Financial Statements

The Bank of New York Company, Inc.

- Consolidated Statements of Condition 24
- Consolidated Statements of Income 25
- Consolidated Statements of Changes in Shareholders' Equity 26
- Consolidated Statements of Changes in Financial Position 27

The Bank of New York

- Consolidated Statements of Condition 28

Notes to Consolidated Financial Statements 29

Opinion of Independent Accountants 37

Discussion and Analysis of the Company's Financial Condition and Results of Its Operations

Selected Financial Data 38
Summary of Results 38
Net Interest Earnings 39
Quarterly Earnings 44
Noninterest Income 45
Noninterest Expense and Income Taxes 45
Liquidity 45
Interest Sensitivity 46
Capital Resources 47
Loans 47
Provision and Allowance for Loan Losses 49
Risk Elements 50
Securities 54
Deposits 55
Short-Term Borrowings 56
Foreign Operations 56
Reporting the Effects of Inflation 58
Financial Ratios 59
Market Price Information 59

The Bank of New York Company, Inc.

Consolidated Statements of Condition

Dollars in thousands	*December 31,* 1984	1983
Assets		
Cash and Due from Banks	$ 871,946	$ 1,031,410
Interest-Bearing Deposits in Banks	1,016,325	989,643
Investment Securities (market value of $1,455,824 in 1984 and $1,229,663 in 1983)	1,480,913	1,281,761
Trading Account Assets	66,595	32,263
Federal Funds Sold and Securities Purchased Under Resale Agreements	419,415	265,167
Loans (less unearned income of $196,450 in 1984 and $130,475 in 1983)	10,303,329	8,140,487
Less: Allowance for Loan Losses	131,250	82,000
Net Loans	10,172,079	8,058,487
Premises and Equipment	121,955	124,640
Due from Customers on Acceptances	659,646	700,576
Accrued Interest Receivable	198,237	151,500
Other Assets	149,644	161,752
Total Assets	$15,156,758	$12,797,199
Liabilities and Shareholders' Equity		
Deposits		
Domestic Offices		
Noninterest-Bearing	$ 2,865,668	$ 2,845,903
Interest-Bearing	4,078,735	4,034,384
Foreign Offices		
Noninterest-Bearing	3,921	4,029
Interest-Bearing	3,579,645	2,675,332
Total Deposits	10,527,969	9,559,648
Federal Funds Purchased and Securities Sold Under Repurchase Agreements	2,166,434	841,120
Commercial Paper	149,965	44,597
Other Short-Term Borrowings	434,750	601,155
Acceptances Outstanding	659,646	700,738
Accrued Taxes and Other Expenses	99,346	126,312
Accrued Interest Payable	124,648	109,954
Other Liabilities	64,535	50,366
Long-Term Debt	169,203	99,306
Total Liabilities	14,396,496	12,133,196
Shareholders' Equity		
Preferred Stock—no par value, authorized 5,000,000 shares Adjustable Rate Cumulative, stated value $50 per share, issued 2,000,000 shares	100,000	100,000
Common Stock—par value $7.50 per share, authorized 50,000,000 shares, issued 16,275,412 in 1984 and 15,315,424 in 1983	122,066	114,866
Capital Surplus	125,649	107,010
Retained Earnings	434,995	365,453
	782,710	687,329
Less: Treasury Stock (488,883 shares in 1984 and 507,986 shares in 1983), at cost	22,448	23,326
Total Shareholders' Equity	760,262	664,003
Total Liabilities and Shareholders' Equity	$15,156,758	$12,797,199

See accompanying Notes to Consolidated Financial Statements

The Bank of New York Company, Inc

Consolidated Statements of Income

Dollars in thousands except per share amounts

	For the years ended December 31,		
	1984	1983	1982
Interest Income			
Interest and Fees on Loans	$1,045,588	$ 882,282	$ 991,266
Interest on Investment Securities			
Taxable	54,323	50,564	46,592
Exempt from Federal Income Taxes	55,442	45,709	48,348
	109,765	96,273	94,940
Interest on Trading Account Assets	2,947	2,158	835
Interest on Deposits in Banks	149,396	133,739	225,905
Interest on Federal Funds Sold and Securities Purchased Under Resale Agreements	25,990	26,275	48,727
Total Interest Income	1,333,686	1,140,727	1,361,673
Interest Expense			
Interest on Deposits	721,582	564,792	757,933
Interest on Federal Funds Purchased and Securities Sold Under Repurchase Agreements	125,095	116,791	165,454
Interest on Commercial Paper	8,617	5,407	3,090
Interest on Other Short-Term Borrowings	54,926	48,825	65,948
Interest on Long-Term Debt	19,040	11,133	11,341
Total Interest Expense	929,260	746,948	1,003,766
Net Interest Income	404,426	393,779	357,907
Provision for Loan Losses	94,003	66,387	49,112
Net Interest Income After Provision for Loan Losses	310,423	327,392	308,795
Noninterest Income			
Trust Fees	152,898	145,087	119,094
Service Charges on Deposit Accounts	20,886	18,315	10,956
Other Charges and Fees	37,996	31,039	21,207
Investment Securities Gains (Losses)	693	(5,489)	(2,652)
Other Operating Income	30,796	9,878	6,599
Total Noninterest Income	243,269	198,830	155,204
Noninterest Expense			
Salaries and Employee Benefits	214,253	207,861	194,235
Net Occupancy Expense of Premises	51,336	49,131	45,000
Furniture and Equipment Expense	30,921	30,126	27,388
Other Operating Expenses	110,465	91,523	92,612
Total Noninterest Expense	406,975	378,641	359,235
Income Before Income Taxes	146,717	147,581	104,764
Income Taxes	39,197	56,969	31,517
Net Income	$ 107,520	$ 90,612	$ 73,247
Net Income Available to Common Shareholders	$ 98,536	$ 83,487	$ 73,247
Per Common Share Data:			
Primary Earnings	$6.40	$5.70	$5.09
Fully Diluted Earnings	5.36	4.72	4.15
Cash Dividends Declared	1.89	1.73½	1.62½
Average Common Shares Outstanding	15,390,131	14,635,773	14,385,312

See accompanying Notes to Consolidated Financial Statements

The Bank of New York Company, Inc.

Consolidated Statements of Changes in Shareholders' Equity

Dollars in thousands

	For the years ended December 31,		
	1984	1983	1982
Preferred Stock			
Balance, January 1	$100,000	$ —	$ —
Issuance of Adjustable Rate Cumulative Preferred Stock (shares: 2,000,000)	—	100,000	—
Balance, December 31 (shares: 2,000,000)	100,000	100,000	—
Common Stock			
Balance, January 1 (shares: 15,315,424 in 1984, 7,616,393 in 1983 and 7,327,808 in 1982)	114,866	57,123	54,959
Two-for-One Stock Split Effected in the Form of a 100% Stock Dividend (shares: 7,302,574)	—	54,769	—
Conversion of Debentures (shares: 153,749 in 1984, 158,880 in 1983 and 174,432 in 1982)	1,153	1,192	1,308
Issuance of Common Shares under Dividend Reinvestment and Benefit Plans (shares: 806,239 in 1984, 237,577 in 1983 and 114,153 in 1982)	6,047	1,782	856
Balance, December 31 (shares: 16,275,412 in 1984, 15,315,424 in 1983 and 7,616,393 in 1982)	122,066	114,866	57,123
Capital Surplus			
Balance, January 1	107,010	153,376	144,430
Two-for-One Stock Split Effected in the Form of a 100% Stock Dividend	—	(54,770)	—
Issuance Costs of Adjustable Rate Cumulative Preferred Stock	(25)	(2,590)	—
Conversion of Debentures	2,099	3,216	5,223
Issuance of Common Shares under Dividend Reinvestment, Compensation and Benefit Plans	16,565	7,778	3,723
Balance, December 31	125,649	107,010	153,376
Retained Earnings			
Balance, January 1	365,453	306,749	256,856
Net Income	107,520	90,612	73,247
Cash Dividends Declared			
Common Stock	(28,978)	(25,468)	(23,354)
Preferred Stock	(9,000)	(6,440)	—
Balance, December 31	434,995	365,453	306,749
Less: Treasury Stock			
Balance, January 1 (shares: 507,986 in 1984, 311,286 in 1983 and 1982)	23,326	12,028	12,028
Issuance of Common Shares under Compensation and Benefit Plans (shares: 19,103 in 1984 and 3,300 in 1983)	(878)	(152)	—
Treasury Stock Acquired (shares: 200,000 in 1983)	—	11,450	—
Balance, December 31 (shares: 488,883 in 1984, 507,986 in 1983 and 311,286 in 1982)	22,448	23,326	12,028
Total Shareholders' Equity, December 31	$760,262	$664,003	$505,220

See accompanying Notes to Consolidated Financial Statements

The Bank of New York Company, Inc.

Consolidated Statements of Changes in Financial Position

Dollars in thousands	*For the years ended December 31,* 1984	1983	1982
Sources of Funds			
Operations			
Net Income	$ 107,520	$ 90,612	$ 73,247
Items Not Using (Providing) Funds During the Period			
Depreciation and Amortization	20,931	16,346	13,670
Provision for Loan Losses	94,003	66,387	49,112
Deferred Income Taxes	15,905	25,821	11,969
Investment Securities Received on the Sale of Branches	(20,000)	—	—
Bond Discount Accrued and Premium Amortized—Net	429	346	(447)
Funds Provided from Operations	218,788	199,512	147,551
Increase in Deposits	968,321	686,470	793,289
Increase in Federal Funds Purchased and Securities Sold Under Repurchase Agreements	1,325,314	—	331,295
Increase in Commercial Paper	105,368	—	93,023
Increase in Other Short-Term Borrowings	—	360,663	—
Increase in Acceptances Outstanding	—	—	263,789
Decrease in Cash and Due from Banks	159,464	—	34,592
Decrease in Interest-Bearing Deposits in Banks	—	708,285	—
Decrease in Investment Securities	—	—	42,252
Decrease in Trading Account Assets	—	17,795	—
Decrease in Federal Funds Sold and Securities Purchased Under Resale Agreements	—	79,271	—
Decrease in Due from Customers on Acceptances	40,930	72,407	—
Issuance of Floating Rate Subordinated Notes Due 1996	75,000	—	—
Issuance of Preferred Stock	—	100,000	—
Issuance of Common Stock from Conversion of Debentures	3,252	4,407	6,531
Proceeds from Sale of Common Stock under Dividend Reinvestment, Compensation and Benefit Plans	22,612	9,560	4,579
Miscellaneous Sources—Net	—	49,365	61,981
Total Sources of Funds	$2,919,049	$2,287,735	$1,778,882
Applications of Funds			
Increase in Cash and Due from Banks	$ —	$ 160,708	$ —
Increase in Interest-Bearing Deposits in Banks	26,685	—	249,707
Increase in Investment Securities	179,581	260,131	—
Increase in Trading Account Assets	34,332	—	46,201
Increase in Federal Funds Sold and Securities Purchased Under Resale Agreements	154,248	—	290,639
Increase in Loans	2,207,595	636,102	564,069
Increase in Due from Customers on Acceptances	—	—	264,357
Decrease in Federal Funds Purchased and Securities Sold Under Repurchase Agreements	—	1,042,554	—
Decrease in Commercial Paper	—	49,929	—
Decrease in Other Short-Term Borrowings	166,405	—	302,162
Decrease in Acceptances Outstanding	41,092	72,251	—
Decrease in Long-Term Debt	5,104	3,094	6,965
Cash Dividends Paid	37,978	31,908	23,354
Purchase of Premises and Equipment	16,410	19,608	31,428
Acquisition of Treasury Stock	—	11,450	—
Miscellaneous Applications—Net	49,619	—	—
Total Applications of Funds	$2,919,049	$2,287,735	$1,778,882

See accompanying Notes to Consolidated Financial Statements

The Bank of New York

Consolidated Statements of Condition

Dollars in thousands

	December 31	
	1984	1983
Assets		
Cash and Due from Banks	$ 866,552	$ 1,030,919
Interest-Bearing Deposits in Banks	1,016,328	989,643
Investment Securities (market value of $1,347,570 in 1984 and $1,168,721 in 1983)	1,385,224	1,224,934
Trading Account Assets	66,595	32,263
Federal Funds Sold and Securities Purchased Under Resale Agreements	419,415	261,400
Federal Funds Sold to Affiliated Bank	788,600	379,841
Loans (less unearned income of $176,760 in 1984 and $120,470 in 1983)	9,219,712	7,707,363
Less Allowance for Loan Losses	129,270	82,000
Net Loans	9,090,442	7,625,363
Premises and Equipment	115,353	120,421
Due from Customers on Acceptances	659,646	700,576
Accrued Interest Receivable	194,939	151,313
Other Assets	113,335	132,633
Total Assets	$14,716,429	$12,649,306
Liabilities and Capital		
Deposits		
Domestic Offices		
Noninterest-Bearing	$ 2,869,873	$ 2,849,191
Interest-Bearing	4,047,105	4,036,107
Foreign Offices		
Noninterest-Bearing	3,921	4,029
Interest-Bearing	3,579,990	2,675,508
Total Deposits	10,500,889	9,564,835
Federal Funds Purchased and Securities Sold Under Repurchase Agreements	2,122,141	894,345
Other Short-Term Borrowings	483,229	581,808
Acceptances Outstanding	659,546	700,738
Accrued Taxes and Other Expenses	84,708	108,460
Accrued Interest Payable	122,076	108,792
Other Liabilities	53,706	44,556
Long-Term Debt	7,070	8,891
Total Liabilities	14,033,465	12,012,425
Capital Accounts		
Common Stock	112,272	112,272
Surplus	158,531	158,531
Undivided Profits	412,161	366,078
Total Capital Accounts	682,964	636,881
Total Liabilities and Capital	$14,716,429	$12,649,306

See accompanying Notes to Consolidated Financial Statements

1. Summary of Significant Accounting and Reporting Policies

The accounting and reporting policies of The Bank of New York Company, Inc. (the Company), a bank holding company, and its subsidiaries, including its principal subsidiary, The Bank of New York (the Bank), conform with generally accepted accounting principles and also with general practice within the banking industry. The following is a summary of the more significant of such policies.

Consolidation

The consolidated financial statements include the accounts of the Company and its subsidiaries. In consolidation, all inter-company accounts and transactions are eliminated.

Securities

Bond investments, which are generally held to maturity, are stated at cost plus discount accrued and less premium amortized. Marketable equity securities are stated at the lower of cost or market value on an aggregate basis. Other equity securities are stated at cost. Trading account securities are carried at market value. Gains and losses on the sale of securities are determined generally on the specific identification method and are included in Noninterest Income.

Loans

Loans are stated at face value, net of unearned income and allowance for loan losses. Unearned income on installment loans is recognized as income over the terms of the loans under the interest method. Interest on other loans is recognized based on the principal amount outstanding.

Lease financings are reported with loans and consist of both direct financing and leveraged leases. Such lease financings are stated net of unearned income and unamortized investment tax credits on leveraged leases. Income is recognized using a method which results in a level rate of return on the net investment in the years when the net investment is positive. The income arising from leveraged leases and direct financing leases is included in Interest and Fees on Loans.

Allowance for Loan Losses

The allowance for loan losses is established through charges to earnings in the form of a provision for loan losses. Loans which are determined to be uncollectable are charged against this allowance account, and subsequent recoveries, if any, are credited to this account. The amount charged to earnings is based on several factors, which include, but are not limited to, analytical reviews of loan loss experience in relation to outstanding loans, a continuing review of problem loans and overall portfolio quality by management and the Audit and Examining Committee of the Company's Board of Directors, regular examinations and appraisals of the loan portfolio conducted by the Bank's internal staff, independent auditors and supervisory authorities, and management's judgment with respect to current and expected economic conditions and their impact on the existing loan portfolio.

Nonaccrual and Reduced Rate Loans

It is the Company's policy to discontinue the accrual of interest on loans (other than installment loans and 1-4 family residential mortgages) on which principal or interest is past due and remains unpaid for 90 days or more and which are not fully collateralized, or whenever there is reasonable doubt that interest and/or principal will be received. Loans which reach nonaccrual status are not restored to accruing status until all delinquent principal and/or interest has been brought current or the loan becomes both fully collateralized and in the process of collection. Generally, any unpaid interest previously accrued is reversed against income, and thereafter interest is recognized only to the extent that cash is received. The Company also modifies the terms of certain loans by reducing the interest rates and extending the maturities to improve ultimate collections of such loans.

Income Taxes

The tax effects of transactions are recognized in the same periods as the related items of income and expense are reported for financial statement purposes, regardless of the period in which such items are recognized for tax purposes. The resulting difference between the tax provision shown in the financial statements and the amount of taxes currently payable constitutes deferred income taxes and is included in Accrued Taxes and Other Expenses, except for deferred investment tax credits relating to leveraged leases. The tax savings resulting from investment tax credits with regard to property purchased by the Company are deferred and amortized to income as reductions of income tax expense.

Per Share Computations	Primary earnings per share is computed by dividing earnings (after deducting dividend requirements on preferred stock) by the weighted average number of common shares outstanding during the period. Fully diluted earnings per share computations give effect to the assumed conversion of the convertible debentures.
Employee Retirement Plan	The Company has a retirement plan covering all eligible employees. The provision for retirement plan costs includes normal service costs, as computed under generally accepted actuarial cost methods and assumptions, plus amortization of unfunded prior service costs over 30 years. The Company's policy is to fund amounts equal to accrued retirement plan costs.
Foreign Currency Translation	Assets and liabilities denominated in foreign currencies are translated principally at current rates of exchange. Income and expense accounts are translated at average rates.
Interest Rate Futures	Gains and losses on interest rate futures contracts entered into for hedging purposes are deferred and amortized over the lives of the hedged assets or liabilities as an adjustment to net interest income. All other gains and losses on futures contracts are recognized currently in operations.
Other	Property held in a fiduciary or agency capacity for customers is not included in the consolidated statements of condition, since such items are not assets of the Company or its subsidiaries. Real estate or other investments acquired in satisfaction of loans are carried at the lower of cost or fair value and are included in Other Assets. Certain 1983 and 1982 information has been reclassified to conform its presentation with the 1984 financial statements.

2. Cash Reserve Requirements

The subsidiary banks of the Company are required to maintain reserve balances with a Federal Reserve Bank under the Federal Reserve Act and Regulation D issued thereunder. Such reserves averaged $233,588,000 and $206,878,000 for the years 1984 and 1983, respectively.

3. Investment Securities

A comparison of the book value and market value of securities follows:

	December 31, 1984		December 31, 1983	
	Book Value	*Market Value*	*Book Value*	*Market Value*
		(In Thousands)		
U.S. Government Obligations	$ 548,624	$ 555,445	$ 520,350	$ 516,982
U.S. Government Agency Obligations	53,850	51,813	53,135	49,093
Obligations of States and Political Subdivisions	783,281	739,681	647,049	597,993
Other Securities	95,158	108,885	61,227	65,595
	$1,480,913	$1,455,824	$1,281,761	$1,229,663

Assets, including securities sold under repurchase agreements, carried at $1,058,253,000 and $949,695,000 at December 31, 1984 and 1983, respectively, were pledged for various purposes as required or permitted by law.

4. Loans

Loans classified by type are as follows:

	December 31, 1984	December 31, 1983
	(In Thousands)	
Real Estate Loans		
Construction and Land Development	$ 898,669	$ 700,246
Secured by 1-4 Family Residential Properties	544,707	536,038
Secured by Multifamily Residential Properties	43,921	52,814
Other, Principally Commercial Mortgages	458,788	393,677
Loans to Domestic Banks	71,110	70,141
Loans for Purchasing or Carrying Securities	1,622,237	800,748
Commercial and Industrial Loans	3,568,150	2,762,358
Loans to Individuals	1,066,267	963,138
Lease Financings	265,167	128,524
Foreign Loans	1,553,079	1,494,319
All Other	407,684	368,959
Total	10,499,779	8,270,962
Less: Unearned Income	196,450	130,475
Allowance for Loan Losses	131,250	82,000
Net Loans	$10,172,079	$8,058,487

Nonaccrual and reduced rate loans outstanding at December 31, 1984, 1983 and 1982 were $291,995,000, $241,056,000 and $213,701,000, respectively. Interest recorded on such loans in 1984, 1983 and 1982 was $8,620,000, $6,714,000 and $8,176,000, respectively. The interest income would have been $37,687,000 in 1984, $33,106,000 in 1983 and $33,146,000 in 1982 if these loans had been current. At December 31, 1984, there were no material commitments to borrowers whose loans were classified as nonaccrual or reduced rate.

The investment in leveraged leases amounted to $194,560,000 at December 31, 1984 and $96,117,000 at December 31, 1983. The deferred taxes arising from leveraged leases of $93,407,000 and $52,987,000 at December 31, 1984 and 1983, respectively, are included in Accrued Taxes and Other Expenses. The net investment in direct financing leases amounted to $9,012,000 at December 31, 1984 and $9,192,000 at December 31, 1983.

5. Allowance for Loan Losses

Transactions in the allowance for loan losses are summarized as follows:

	1984	1983	1982
		(In Thousands)	
Balance, January 1	$ 82,000	$64,100	$44,300
Loans Charged Off	(52,430)	(54,603)	(32,894)
Recoveries	7,677	6,116	3,582
Net Loans Charged Off	(44,753)	(48,487)	(29,312)
Provision Charged to Income	94,003	66,387	49,112
Balance, December 31	$131,250	$82,000	$64,100

6. Long-Term Debt

The following is a summary of long term debt

	December 31 1984	December 31 1983
	(In Thousands)	
Floating Rate Subordinated Notes Due 1996	$ 75,000	$ —
12% Convertible Subordinated Debentures Due 2006	73,352	75,000
6¼% Convertible Debentures Due 1994	12,052	13,683
Other	8,799	10,623
	$169,203	$99,306

On January 12, 1984, The Bank of New York Overseas Finance N.V. (Finance), a wholly owned subsidiary of the Company, issued $75,000,000 of floating rate subordinated notes due 1996. The payment of principal and interest is unconditionally guaranteed, on a subordinated basis, by the Company. The interest rate is adjusted quarterly based upon prevailing rates of three month Eurodollar deposits, subject to a minimum of 5.25% per annum. The notes may be redeemed, at the option of Finance, at their principal amount on or after the interest payment date occurring in January 1985. Because the Company has committed to issue permanent equity capital in the amount of $25,000,000 in each four year period during which the notes are outstanding, the notes are considered primary capital of the Company for bank regulatory purposes. As of December 31, 1984 the Company attributed $5,700,000 of equity capital issued during 1984 to the first four-year commitment.

On January 16, 1985, the 12% and 6¼% convertible debentures were called for redemption on February 15, 1985 at 109.6% and 107.5%, respectively, of principal amount. The holders of the debentures may elect before February 16, 1985 to convert them into the Company's common stock at a price of $24.625 and $18.75 per share, respectively.

7. Preferred Stock

The Company has outstanding 2,000,000 shares of adjustable rate cumulative preferred stock at $50 per share. Holders of preferred stock have cumulative dividend rights in preference to holders of common stock. The dividend rate is adjusted quarterly based upon prevailing interest rates. These quarterly rates are subject to minimum and maximum limitations of 6.00% and 12.00% per annum. The stock may be redeemed at the option of the Company on or after February 15, 1988 and prior to February 15, 1993 at a price of $51.50 per share and thereafter at $50 per share, plus cumulative and unpaid dividends to the redemption date. The liquidation preference is $50 per share, plus cumulative and unpaid dividends.

8. Incentive Stock Option Plan

In April 1984, the shareholders approved the 1984 Incentive Stock Option Plan (the Plan), which provides for the issuance of stock options to officers and key employees of the Company and its subsidiaries. The Plan provides for the granting of options for 750,000 shares of the Company's common stock at the fair market value at the time the options are granted. Each option granted under the Plan is exercisable commencing at least one year following the date of grant, and in no event later than ten years from the date of grant. During 1984, 166,125 options were granted at an exercise price of $28.125 per share, of which 5,275 were cancelled. At December 31, 1984, 583,875 shares of common stock were available for future issuance under the Plan.

9. Income Taxes

Income taxes included in the consolidated statements of income are as follows:

	1984			1983*			1982*		
	Current	*Deferred*	*Total*	*Current*	*Deferred*	*Total*	*Current*	*Deferred*	*Total*
					(In Thousands)				
Federal	$ 3,365	$18,340	$21,705	$ 8,024	$19,679	$27,703	$ 3,000	$12,667	$15,667
Foreign	4,228	–	4,228	3,811	–	3,811	946	–	946
State and local	15,699	(2,435)	13,264	19,313	6,142	25,455	15,602	(698)	14,904
	$23,292	$15,905	$39,197	$31,148	$25,821	$56,969	$19,548	$11,969	$31,517

*Restated to reflect reclassification of current and deferred taxes upon filing of tax returns

The tax (benefit) provision for investment securities gains and losses is included in the above totals and amounted to $413,000 in 1984, $(3,320,000) in 1983 and $(1,566,000) in 1982.

For the computation of taxes, the components of income before taxes are as follows:

	1984	1983	1982
		(In Thousands)	
Domestic	$124,304	$128,929	$ 86,440
Foreign	22,413	18,652	18,324
	$146,717	$147,581	$104,764

The sources of deferred income taxes are as follows:

	1984	1983	1982
		(In Thousands)	
Investment tax credits	$11,019	$ 4,528	$ 10,101
Lease financings	42,591	20,039	11,265
Provision for loan losses	(27,823)	(10,062)	(11,495)
Other	(9,882)	11,316	2,098
	$15,905	$25,821	$ 11,969

The consolidated provision for federal income taxes is less than the amounts computed by applying the federal income tax rate of 46% to income before taxes, due to the following:

	1984		1983		1982	
	Amount	*% of Pre Tax Income*	*Amount*	*% of Pre Tax Income*	*Amount*	*% of Pre Tax Income*
			(In Thousands)			
Computed provision using 46%	$67,490	46.0%	$67,887	46.0%	$48,191	46.0%
Increase due to state and local income taxes net of federal income tax benefit	7,163	4.9	13,746	9.3	8,048	7.7
Decrease due to tax-exempt interest	(24,662)	(16.8)	(21,102)	(14.3)	(22,604)	(21.6)
Decrease due to investment tax credits	(5,491)	(3.7)	(2,637)	(1.8)	(1,915)	(1.8)
Decrease due to capital gains on branch sales	(4,397)	(3.0)	(613)	(.4)	(252)	(.2)
Other	(906)	(.7)	(312)	(.2)	49	–
	$39,197	26.7%	$56,969	38.6%	$31,517	30.1%

10. Employee Retirement Plan

The Company has a retirement plan covering substantially all employees. The total provision for retirement plan costs was $6,363,000 for 1984, $7,091,000 for 1983 and $6,771,000 for 1982. A comparison of accumulated plan benefits and plan net assets for the Company's pension plan is presented below:

	January 1, 1984	January 1, 1983
	(In Thousands)	
Actuarial present value of accumulated plan benefits:		
Vested	$ 64,104	$ 56,402
Nonvested	4,152	2,670
	$ 68,256	$ 59,072
Net assets available for benefits	$121,241	$109,061

The assumed rate of return used in determining the actuarial present values of vested and nonvested accumulated plan benefits was 9½% and 10% for January 1, 1984 and 1983, respectively.

11. Foreign Activities

The Company's foreign activities, as shown in the table that follows, consist of banking services provided to customers domiciled outside of the United States. Assets and liabilities, as well as revenues and expenses, applicable to specific customers are directly attributed upon this basis. Complete attribution is not possible due to the integration of domestic and foreign activities, accordingly, certain allocations have been made to present this data. The major allocations are the charges for funds employed, which are based upon average costs of interest-bearing funds, and indirect expenses, which are distributed through the use of various internal estimates. These allocations do not have any material impact on any one geographical region, as they have been allocated in relationship to the total operating income of each region. A provision for loan losses is not reflected in foreign income except to recognize direct net charge-offs.

Geographical Distribution

The Company's foreign activities by significant geographical region are as follows:

	Total Revenues	*Income Before Income Taxes*	*Net Income*	As of December 31 *Total Assets*	As of December 31 *Total Liabilities*
		(In Thousands)		*(In Thousands)*	
	For the Year Ended December 31, 1984			1984	1984
Europe	$181,258	$11,465	$ 6,191	$1,227,967	$1,357,268
Middle East and Africa	10,045	677	366	93,499	52,522
Asia and Pacific	83,233	5,558	3,001	730,424	505,220
Western Hemisphere	142,390	8,899	4,805	985,893	724,245
Total	$416,926	$26,599	$14,363	$3,037,783	$2,639,255
	For the Year Ended December 31, 1983			1983	1983
Europe	$214,285	$10,724	$ 5,791	$1,472,760	$1,384,005
Middle East and Africa	10,838	624	337	135,076	38,910
Asia and Pacific	50,584	2,430	1,312	478,020	345,725
Western Hemisphere	108,508	5,633	3,042	911,245	956,790
Total	$384,215	$19,411	$10,482	$2,997,101	$2,725,430
	For the Year Ended December 31, 1982			1982	1982
Europe	$232,509	$10,109	$ 5,458	$1,884,924	$1,323,383
Middle East and Africa	11,779	315	170	155,207	6,200
Asia and Pacific	62,476	2,864	1,547	546,449	505,831
Western Hemisphere	172,789	7,709	4,163	1,010,679	786,797
Total	$479,553	$20,997	$11,338	$3,597,259	$2,622,211

Foreign Exchange

Foreign exchange activities arise from servicing customers' needs for foreign exchange and from the Bank's engaging in spot and forward transactions to utilize and hedge foreign currency positions. Net foreign currency exchange adjustments, resulting from foreign exchange transactions and translation of foreign balances, were gains of $2,798,000, $1,076,000 and $2,402,000 in 1984, 1983 and 1982, respectively, which were included in Other Operating Income.

12. Parent Company Financial Information

The condensed financial statements of the Parent Company are as follows:

Balance Sheets

	December 31, 1984	December 31, 1983
	(In Thousands)	
Assets		
Cash	$ 91	$ 65
Interest-Bearing Deposits in Banks	422	—
Investment Securities, at cost	6,32[illegible]	1,220
Investment in Subsidiaries, at equity		
Bank	841,153	720,681
Other	22,817	19,227
	863,970	739,908
Due from Subsidiaries		
Bank	99,851	29,681
Other	79,805	43,104
	179,656	72,785
Other Assets	23,119	10,370
Total Assets	$1,073,581	$824,348
Liabilities and Shareholders' Equity		
Commercial Paper	$ 149,965	$ 44,597
Due to Nonbank Subsidiary	76,050	—
Other Liabilities	1,900	27,065
Long-Term Debt	85,404	88,683
Total Liabilities	313,319	160,345
Shareholders' Equity*	760,262	664,003
Total Liabilities and Shareholders' Equity	$1,073,581	$824,348

See Consolidated Statements of Changes in Shareholders' Equity on page 26

Statements of Income

	For the years ended December 31, 1984	1983	1982
		(In Thousands)	
Operating Income			
Dividends from Bank Subsidiary	$ 51,800	$40,000	$34,500
Interest on Due from Subsidiaries			
Bank	6,317	5,302	1,530
Other	5,490	5,012	1,980
Other Interest and Dividends	267	1,291	1,760
Other Operating Income	184	786	834
Total	64,058	52,391	40,604
Operating Expenses			
Interest on Commercial Paper	8,617	5,407	3,090
Interest on Due to Nonbank Subsidiary	8,502	—	—
Interest on Long-Term Debt	9,679	9,951	10,272
Other Operating Expenses	4,913	1,763	1,781
Total	31,711	17,121	15,143
Income Before Income Taxes and Equity in Undistributed Earnings of Subsidiaries	32,347	35,270	25,461
Income Tax Benefit	10,596	2,964	4,913
Income Before Equity in Undistributed Earnings of Subsidiaries	42,943	38,234	30,374
Equity in Undistributed Earnings of Subsidiaries			
Bank	60,974	48,547	40,457
Other	3,603	3,831	2,416
	64,577	52,378	42,873
Net Income	$107,520	$90,612	$73,247

Statements of Changes in Financial Position

	For the years ended December 31, 1984	1983	1982
	(In Thousands)		
Sources of Funds			
Operations			
Net Income	$107,520	$ 90,612	$ 73,247
Less: Equity in Undistributed Earnings of Subsidiaries	(64,577)	(52,378)	(42,873)
Funds Provided from Operations	42,943	38,234	30,374
Decrease in Other Investments	—	23,159	—
Decrease in Due from/to Subsidiaries	—	38,426	—
Increase in Commercial Paper	105,368	—	93,023
Issuance of Preferred Stock	—	100,000	—
Issuance of Common Stock from Conversion of Debentures	3,252	4,407	6,531
Proceeds from Sale of Common Stock under Dividend Reinvestment, Compensation and Benefit Plans	22,612	9,560	4,579
Miscellaneous Sources—Net	—	—	15,256
Total Sources of Funds	$174,175	$213,786	$149,763
Applications of Funds			
Increase in Other Investments	$ 5,525	$ —	$ 9,812
Capital Contribution to Subsidiaries	59,800	117,449	—
Increase in Due from/to Subsidiaries	30,821	—	110,050
Decrease in Commercial Paper	—	49,929	—
Decrease in Long-Term Debt	3,279	4,422	6,547
Cash Dividends Paid	37,978	31,908	23,354
Miscellaneous Applications—Net	36,772	10,078	—
Total Applications of Funds	$174,175	$213,786	$149,763

There are various restrictions which limit the ability of subsidiary banks to transfer funds in the form of cash dividends, loans or advances to the Parent Company. Under federal law, the approval of the primary regulator is required if dividends declared by the Bank and other insured bank subsidiaries in any year exceed the net profits of that year (as defined) combined with the retained net profits for the two preceding years. During 1985, these companies could, without approval of appropriate regulatory authorities, declare dividends amounting to $120,659,000 plus their net profits in 1985.

In addition, the subsidiary banks are subject to restrictions under Section 23A of the Federal Reserve Act. These restrictions limit the transfer of funds to the Parent and certain other affiliates, as defined, in the form of loans or extensions of credit, investments and purchases of assets. Such transfers by a bank to the Parent or any such affiliate are limited in amount to 10% of the bank's capital and surplus and, in the aggregate, to 20% of the bank's capital and surplus. These transfers are also subject to various collateral requirements.

13. Lease Commitments

Net rent expense for premises and equipment was $31,058,000 in 1984, $30,565,000 in 1983 and $27,404,000 in 1982.

At December 31, 1984, the Company and its subsidiaries were obligated under various noncancelable lease agreements, certain of which provide for additional rents based upon real estate taxes, insurance, and maintenance and for various renewal options. The minimum rental commitments under noncancelable operating leases for premises and equipment having a term of more than one year from December 31 are as follows:

Year ending December 31,	*(In Thousands)*
1985	$ 20,612
1986	18,318
1987	16,268
1988	14,349
1989	13,644
Subsequent to 1989	93,627
Total minimum lease payments	$176,818

14. Commitments and Contingent Liabilities

In the normal course of business, there are outstanding various commitments and contingent liabilities, such as commitments to extend credit, the purchase and sale of foreign exchange under futures contracts and interest rate futures contracts, and guarantees, which are not reflected in the accompanying consolidated financial statements. Management does not expect any material losses to result from these transactions. Outstanding standby letters of credit, net of participations, were $1,085,151,000 and $791,007,000 at December 31, 1984 and December 31, 1983, respectively. Standby letters of credit included in the above that were fully collateralized at December 31, 1984 and December 31, 1983 totaled $309,290,000 and $256,631,000, respectively. Participations conveyed to others amounted to $75,666,000 at December 31, 1984 and none at December 31, 1983.

There are also various claims pending against the Company and the Bank. In the opinion of management, liabilities arising from such claims, if any, would not have a material effect upon the Company's consolidated financial statements.

15. Reporting the Effects of Inflation (Unaudited)

A five-year comparison of selected supplementary financial data, adjusted for the effects of changing prices, and discussion thereof are found in the Discussion and Analysis section of this report on page 58.

Opinion of Independent Accountants

Deloitte
Haskins + Sells

The Shareholders and Board of Directors of
The Bank of New York Company, Inc.:

We have examined the consolidated statements of condition of The Bank of New York Company, Inc. and its subsidiaries as of December 31, 1984 and 1983 and the related consolidated statements of income, changes in shareholders' equity and changes in financial position for each of the three years in the period ended December 31, 1984 and the consolidated statements of condition of The Bank of New York and its subsidiaries as of December 31, 1984 and 1983, appearing herein on pages 24 through 37. Our examinations were made in accordance with generally accepted auditing standards and, accordingly, included such tests of the accounting records and such other auditing procedures as we considered necessary in the circumstances.

In our opinion, such financial statements present fairly the financial position of The Bank of New York Company, Inc. and its subsidiaries at December 31, 1984 and 1983 and the results of their operations and the changes in their financial position for each of the three years in the period ended December 31, 1984 and the financial position of The Bank of New York and its subsidiaries at December 31, 1984 and 1983, in conformity with generally accepted accounting principles applied on a consistent basis.

Deloitte Haskins + Sells

New York, N.Y.
January 28, 1985

Discussion and Analysis of the Company's Financial Condition and Results of Its Operations

This section presents management's discussion and analysis of the financial condition and results of operations of The Bank of New York Company, Inc. (the Company) and its subsidiaries, including its principal subsidiary, The Bank of New York (the Bank). The discussion and analysis should be read in conjunction with the financial statements and supplemental financial data contained elsewhere in this report.

Selected Financial Data *(In Millions except per share amounts)*

	1984	1983	1982	1981	1980	1979	1984 Compared with 1983 Increase (Decrease)	1983 Compared with 1982 Increase (Decrease)
Net Interest Income	$404.4	$393.8	$357.9	$277.5	$223.3	$198.1	3%	10%
Provision for Loan Losses	94.0	66.4	49.1	19.5	16.8	24.8	42	35
Noninterest Income	243.3	198.8	155.2	112.7	83.1	64.9	22	28
Noninterest Expense	407.0	378.6	359.2	293.2	223.6	180.0	7	5
Applicable Income Taxes	39.2	57.0	31.5	19.2	17.2	14.6	(31)	81
Income Before Extraordinary Charge	107.5	90.6	73.2	58.2	48.8	43.6	19	24
Net Income	107.5	90.6	73.2	58.2	42.6	43.6	19	24
Net Income Available to Common Shareholders	98.5	83.5	73.2	58.2	42.6	43.6	18	14
Per Common Share Data:								
Primary Earnings:								
Income Before Extraordinary Charge	$6.40	$5.70	$5.09	$4.25	$3.74	$3.53	12%	12%
Net Income	6.40	5.70	5.09	4.25	3.27	3.53	12	12
Fully Diluted Earnings:								
Income Before Extraordinary Charge	5.36	4.72	4.15	3.80	3.25	2.94	14	14
Net Income	5.36	4.72	4.15	3.80	2.85	2.94	14	14
Cash Dividends Declared	1.89	1.73½	1.62½	1.52½	1.39½	1.25½	9	7
Average Total Assets	$13,380.1	$12,698.9	$12,243.8	$11,233.0	$10,077.5	$8,437.0	5%	4%
Average Long-Term Debt	169.4	101.8	105.8	45.0	48.0	58.5	66	(4)
Average Common Shareholders' Equity	627.4	540.9	478.8	418.1	384.7	341.4	16	13

Summary of Results

The Bank of New York Company, Inc. reported record earnings in 1984 for the seventh consecutive year, posting net income of $107.5 million, up 18.7% from 1983's net income of $90.6 million. Fully diluted earnings per share, which assume full conversion of the Company's debentures, rose 13.6% to $5.36 in 1984 from $4.72 in 1983.

During 1984, the sustained economic recovery continued to produce a relatively stable interest rate environment with low inflation, while favorable debt restructuring agreements improved the economic condition of many developing countries. Offsetting these positive factors were the uncertainty of the effect of large present and future budget deficits, which kept real interest rates at high levels, and earnings problems at a few bank holding companies that increased the regulatory pressure on all banking institutions to maintain higher capital ratios. The Company was able to increase its primary capital by $214.8 million, or 28.8%, to $960.8 million from $746.0 million in 1983. At the same time the Company's total asset base rose by 18.4% to almost $15.2 billion from $12.8 billion at year-end 1983. As a result, the Company's year-end primary capital ratio increased to 6.34% from 5.83% in 1983. Net loan charge-offs declined during 1984 while management continued its prudent policy of increasing the loan loss provision, which resulted in a substantially improved allowance for loan losses. This allowance, maintained to provide protection for shareholders against any loan charge-offs, stood at 1.27% of total loans at year-end 1984, up from 1.01% in 1983 and .85% in 1982.

The Company's strong 1984 performance was balanced and diverse. Strong loan growth, particularly domestic, led the Company to report record net interest income. Noninterest revenues also increased to record levels, reflecting primarily higher trust and other fees and capital gains from the sale of 36 bank branches. Noninterest expense growth was held to 7.5%, the result of management's aggressive cost consciousness program and higher employee productivity. The Company's performance indices for the year highlighted these achievements: return on average assets improved to .80% in 1984 from .71% in 1983 and .60% in 1982, while return on average common shareholders' equity increased to 15.71% from 15.43% in 1983 and 15.30% in 1982. The Company's return on assets and common equity ratios are among the highest of all New York money-center bank holding companies.

During the fourth quarter of 1984, the Company again reported record earnings of $28.4 million versus $23.5 million in 1983, an increase of 21.1%. Many of the same factors which resulted in a strong annual performance also resulted in the gain during the quarter, as discussed more fully in the section on Quarterly Earnings. Return on average assets was .81%, and return on average common shareholders' equity was 15.28%.

Net Interest Earnings

Net interest earnings, the Company's primary source of consolidated income, is the difference between interest and fees derived from loans, investments and other interest-earning assets and interest paid on liabilities. For analytical purposes, those forms of income that are tax-exempt are adjusted to a fully taxable basis in order to present income and yields on all earning assets on a comparable basis. In 1984, a taxable equivalent adjustment of $71.7 million at a combined incremental rate of 117.4% was added to total interest income. The Company achieved record net interest earnings of $476.1 million in 1984, an increase of 6.6% over 1983. The increase of $29.6 million is primarily attributable to the growth of earning assets, particularly loans.

Earning assets of $11.5 billion were up $884.5 million from 1983, an increase of 8.3%. Gross revenues generated by all interest-earning assets in 1984 reached $1.4 billion at an average rate of 12.23%, with the increase from 1983 attributable to both larger volume and higher rates. Loans, the largest component of earning assets, showed significant volume increases in both domestic and foreign categories. Foreign loans are defined as loans generated by the Company's foreign branches and international banking subsidiaries and facility. Loans remain the most profitable source of income and, as such, are carefully managed with respect to the Company's interest sensitivity position. The average balances of both interest-bearing deposits in banks and federal funds sold and securities purchased under resale agreements continued their decline from 1982, as narrower spreads reduced their overall attractiveness. This decrease is indicative of management's capacity to effect changes in the Company's asset mix in order to capitalize on more desirable yields on other interest-earning assets.

The significant increase in interest expense of $182.3 million, to $929.3 million, in 1984 was also attributable to both increased volume and higher rates. The level of funds available without interest cost, chiefly demand deposits, is an important factor in the growth of net interest earnings. These sources of funds have declined due to the introduction of more attractive consumer products, such as the short-term money market accounts first offered in December 1982. The Company has seen these accounts grow significantly since 1982 to $1.6 billion on average during 1984. On December 31, 1984 the balance in these accounts amounted to $1.8 billion.

Total average interest-bearing deposits increased 11.4% to $7.2 billion in 1984, with the largest components of the increase being the growth in money market deposits mentioned above, an increase in large certificates of deposit of $298.4 million, and an increase in deposits in foreign offices of $160.2 million. Because of the number and diversity of depositors and deposit instruments, these sources of funds have remained an effective and stable funding mechanism for the Company.

As a result of the faster rate of growth in the volume of and the rates paid on interest-bearing liabilities, the Company's interest rate spread, the difference between the average rate earned on all earning assets and the average cost of all interest-bearing sources of funds, declined from 2.37% in 1983 to 2.13% in 1984. Because of the increased value of noninterest-bearing sources of funds (including demand deposits, other liabilities and capital) in a higher interest rate environment, the Company's net yield decreased only from 4.21% to 4.14%.

Average Balances and Rates on a Taxable Equivalent Basis *(In Thousands)*

	1984			1983			1982		
	Average Balance	*Interest*	*Average Rate*	*Average Balance*	*Interest*	*Average Rate*	*Average Balance*	*Interest*	*Average Rate*
Assets									
Interest-Bearing Deposits in Banks	$ 1,177,594	$ 149,396	12.69%	$ 1,224,341	$ 133,739	10.92%	$ 1,553,778	$ 225,905	14.54%
Federal Funds Sold and Securities Purchased Under Resale Agreements	242,162	25,990	10.73	280,206	26,275	9.38	381,145	48,727	12.78
Loans*									
Domestic Offices	6,106,000	734,763	12.03	5,557,768	629,390	11.32	5,286,214	719,575	13.61
Foreign Offices	2,676,494	317,696	11.87	2,383,242	255,445	10.72	1,956,427	273,245	13.97
Total Loans	8,782,494	1,052,459	11.98	7,941,010	884,835	11.14	7,242,641	992,820	13.71
U.S. Government Obligations	413,507	43,195	10.45	411,885	41,815	10.15	316,142	35,767	11.31
U.S. Government Agency Obligations	47,650	3,562	7.48	53,663	4,065	7.58	71,478	5,338	7.47
Obligations of States and Political Subdivisions	701,214	111,389	15.89	609,587	93,392	15.32	620,086	100,586	16.22
Other Securities, including Trading Account Securities	124,752	19,402	15.55	84,151	9,366	11.13	53,772	7,045	13.10
Total Securities	1,287,123	177,548	13.79	1,159,286	148,638	12.82	1,061,478	148,736	14.01
Total Interest-Earning Assets	11,489,373	1,405,393	12.23%	10,604,843	1,193,487	11.25%	10,239,042	1,416,188	13.83%
Allowance for Loan Losses	(108,294)			(78,720)			(54,857)		
Cash and Due from Banks	865,936			1,037,505			976,237		
Other Assets	1,133,077			1,135,227			1,083,338		
Total Assets	$13,380,092			$12,698,855			$12,243,760		
Liabilities and Shareholders' Equity									
Deposits									
Money Market Accounts	$ 1,613,550	149,063	9.24%	$ 1,223,663	102,427	8.37%	$ 5,332	522	9.79%
Savings	752,067	38,175	5.08	878,958	44,690	5.08	977,753	49,948	5.11
Certificates of Deposit of $100,000 or More	684,857	72,650	10.61	386,410	35,857	9.28	761,319	98,411	12.93
Other Time Deposits	1,142,490	133,003	11.64	1,133,342	107,665	9.50	1,556,465	188,183	12.09
Foreign Offices	2,968,702	328,691	11.07	2,808,472	274,153	9.76	3,135,390	420,869	13.42
Total Deposits	7,161,666	721,582	10.08	6,430,845	564,792	8.78	6,436,259	757,933	11.78
Federal Funds Purchased and Securities Sold Under Repurchase Agreements	1,262,195	125,095	9.91	1,316,304	116,791	8.87	1,363,430	165,454	12.14
Commercial Paper	80,105	8,617	10.76	58,293	5,407	9.28	27,145	3,090	11.38
Other Short-Term Borrowings	523,412	54,926	10.49	503,471	48,825	9.70	462,307	65,948	14.26
Long-Term Debt	169,379	19,040	11.24	101,824	11,133	10.93	105,841	11,341	10.72
Total Interest-Bearing Liabilities	9,196,757	929,260	10.10	8,410,737	746,948	8.88	8,394,982	1,003,766	11.96
Noninterest-Bearing Deposits	2,517,460			2,743,292			2,489,199		
Other Liabilities	938,458			918,405			880,802		
Preferred Stock	100,000			85,480			—		
Common Shareholders' Equity	627,417			540,941			478,777		
Total Liabilities and Shareholders' Equity	$13,380,092			$12,698,855			$12,243,760		
Net Interest Earnings and Interest Rate Spread		$ 476,133	2.13%		$ 446,539	2.37%		$ 412,422	1.87%
Net Yield on Interest-Earning Assets			4.14%			4.21%			4.03%

*Loan interest includes fees on loans of $37,498,000 in 1984, $36,950,000 in 1983 and $32,056,000 in 1982.

For purposes of this schedule, nonaccrual loans are included in the average loan balance, and any associated income is recognized on a cash basis.

Rate/Volume Analysis on a Taxable Equivalent Basis *(In Thousands)*

	1984 vs. 1983			1983 vs. 1982		
	Increase (decrease) due to change in			*Increase (decrease) due to change in*		
	Average Balance	*Average Rate*	*Total Increase (Decrease)*	*Average Balance*	*Average Rate*	*Total Increase (Decrease)*
Interest Income						
Interest-Bearing Deposits in Banks	$ (5,931)	$ 21,588	$ 15,657	$ (42,415)	$ (49,751)	$ (92,166)
Federal Funds Sold and Securities Purchased Under Resale Agreements	(4,083)	3,798	(285)	(11,190)	(11,262)	(22,452)
Loans						
Domestic Offices	64,462	40,911	105,373	36,965	(127,150)	(90,185)
Foreign Offices	33,235	29,016	62,251	59,611	(77,411)	(17,800)
Total Loans	97,697	69,927	167,624	96,576	(204,561)	(107,985)
U.S. Government Obligations	165	1,215	1,380	10,832	(4,784)	6,048
U.S. Government Agency Obligations	(450)	(53)	(503)	(1,349)	76	(1,273)
Obligations of States and Political Subdivisions	14,453	3,544	17,997	(1,681)	(5,513)	(7,194)
Other Securities, including Trading Account Securities	5,504	4,532	10,036	3,980	(1,659)	2,321
Total Interest Income	107,355	104,551	211,906	54,753	(277,454)	(222,701)
Interest Expense						
Deposits						
Money Market Accounts	35,188	11,448	46,636	119,274	(17,369)	101,905
Savings	(6,441)	(74)	(6,515)	(5,024)	(234)	(5,258)
Certificates of Deposit of $100,000 or More	31,040	5,753	36,793	(39,770)	(22,784)	(62,554)
Other Time Deposits	876	24,462	25,338	(45,027)	(35,491)	(80,518)
Foreign Offices	16,267	38,271	54,538	(40,573)	(106,143)	(146,716)
Total Deposits	76,930	79,860	156,790	(11,120)	(182,021)	(193,141)
Federal Funds Purchased and Securities Sold Under Repurchase Agreements	(5,363)	13,667	8,304	(5,544)	(43,119)	(48,663)
Commercial Paper	2,250	960	3,210	3,546	(1,229)	2,317
Other Short-Term Borrowings	1,986	4,115	6,101	5,872	(22,995)	(17,123)
Long-Term Debt	7,585	322	7,907	(439)	231	(208)
Total Interest Expense	83,388	98,924	182,312	(7,685)	(249,133)	(256,818)
Change in Net Interest Income	$ 23,967	$ 5,627	$ 29,594	$ 62,438	$ (28,321)	$ 34,117
Percentage Change in Net Interest Income over Prior Period			6.6%			8.3%

For purposes of this table, changes which are not solely due to balance changes or solely due to rate changes are allocated to such categories on the basis of the respective percentage changes in average balances and average rates.

Average Balances and Rates on a Taxable Equivalent Basis by Quarter *(In Thousands)*

	1984 Fourth Quarter			1984 Third Quarter		
	Average Balance	*Interest*	*Average Rate*	*Average Balance*	*Interest*	*Average Rate*
Assets						
Interest-Bearing Deposits in Banks	$ 1,055,167	$ 35,363	13.33%	$ 1,382,523	$ 48,009	13.81%
Federal Funds Sold and Securities Purchased Under Resale Agreements	204,797	5,010	9.73	205,744	6,423	12.42
Loans						
Domestic Offices	6,504,820	191,166	11.69	5,983,158	190,273	12.65
Foreign Offices	2,799,647	86,722	12.32	2,667,086	82,717	12.34
Total Loans	9,304,467	277,888	11.88	8,650,244	272,990	12.55
U.S. Government Obligations	483,264	13,176	10.85	384,208	10,237	10.60
U.S. Government Agency Obligations	46,896	851	7.22	45,046	848	7.49
Obligations of States and Political Subdivisions	749,772	29,960	15.90	714,083	28,700	15.99
Other Securities, including Trading Account Securities	144,507	6,669	18.36	143,350	5,901	16.38
Total Securities	1,424,439	50,656	14.15	1,286,687	45,686	14.13
Total Interest-Earning Assets	11,988,870	368,917	12.24%	11,525,198	373,108	12.88%
Allowance for Loan Losses	(122,896)			(118,404)		
Cash and Due from Banks	875,255			802,970		
Other Assets	1,203,226			1,192,759		
Total Assets	$13,944,455			$13,402,523		
Liabilities and Shareholders' Equity						
Deposits						
Money Market Accounts	$ 1,711,895	38,697	8.99%	$ 1,593,073	39,941	9.97%
Savings	691,617	8,811	5.07	702,003	8,940	5.07
Certificates of Deposit of $100,000 or More	570,709	15,155	10.56	928,530	26,506	11.36
Other Time Deposits	1,144,208	34,019	11.83	1,121,497	31,469	11.16
Foreign Offices	3,094,138	86,288	11.09	3,083,665	94,235	12.16
Total Deposits	7,212,567	182,970	10.09	7,428,768	201,091	10.77
Federal Funds Purchased and Securities Sold Under Repurchase Agreements	1,888,827	43,392	9.14	1,062,299	29,805	11.16
Commercial Paper	94,585	2,416	10.15	116,411	3,403	11.63
Other Short-Term Borrowings	337,497	8,996	10.60	544,831	15,179	11.08
Long-Term Debt	170,258	4,861	11.36	171,173	5,023	11.67
Total Interest-Bearing Liabilities	9,703,734	242,635	9.95	9,323,482	254,501	10.86
Noninterest-Bearing Deposits	2,499,114			2,354,138		
Other Liabilities	959,295			996,895		
Preferred Stock	100,000			100,000		
Common Shareholders' Equity	682,312			628,008		
Total Liabilities and Shareholders' Equity	$13,944,455			$13,402,523		
Net Interest Earnings and Interest Rate Spread		$126,282	2.29%		$118,607	2.02%
Net Yield on Interest-Earning Assets			4.19%			4.09%

1984 Second Quarter			1984 First Quarter			1983 Fourth Quarter		
Average Balance	*Interest*	*Average Rate*	*Average Balance*	*Interest*	*Average Rate*	*Average Balance*	*Interest*	*Average Rate*
$ 1,278,105	$ 37,684	11.86%	$ 993,675	$ 28,340	11.47%	$ 1,079,378	$ 29,969	11.02%
258,412	6,760	10.52	300,506	7,797	10.44	339,400	8,233	9.62
6,056,979	183,253	12.17	5,876,011	170,071	11.64	5,635,374	161,579	11.38
2,679,168	77,624	11.65	2,558,824	70,633	11.10	2,460,026	71,353	11.51
8,736,147	260,877	12.01	8,434,835	240,704	11.48	8,095,400	232,932	11.42
373,256	9,687	10.44	412,856	10,095	9.83	509,664	12,465	9.70
45,786	862	7.57	52,908	1,001	7.61	53,132	1,006	7.51
683,956	27,108	15.94	656,370	25,621	15.70	628,566	23,952	15.12
113,952	3,952	13.95	96,778	2,880	11.97	93,397	2,747	11.67
1,216,950	41,609	13.75	1,218,912	39,597	13.07	1,284,759	40,170	12.40
11,489,614	346,930	12.14%	10,947,928	316,438	11.63%	10,798,937	311,304	11.44%
(104,617)			(86,987)			(85,260)		
904,632			881,476			923,685		
1,059,307			1,075,589			1,154,159		
$13,348,936			$12,818,006			$12,791,521		
$ 1,611,818	37,084	9.25%	$ 1,536,558	33,341	8.73%	$ 1,425,926	30,634	8.52%
793,808	9,966	5.05	822,054	10,458	5.12	853,6[illegible]	10,976	5.10
717,301	18,491	10.37	521,465	12,498	9.64	565,673	13,476	9.45
1,176,946	37,452	12.80	1,127,520	30,063	10.72	1,207,085	29,732	9.77
2,923,501	78,058	10.74	2,770,863	70,110	10.18	2,706,782	67,706	9.92
7,223,374	181,051	10.08	6,778,460	156,470	9.28	6,759,163	152,524	8.95
1,128,116	28,929	10.31	964,848	22,969	9.57	942,956	21,754	9.15
66,038	1,757	10.70	42,828	1,041	9.78	20,046	481	9.52
619,188	16,256	10.56	593,940	14,495	9.82	626,799	16,351	10.35
171,585	4,785	11.22	164,471	4,371	10.69	99,931	2,778	11.03
9,208,301	232,778	10.17	8,544,547	199,346	9.38	8,448,895	193,888	9.10
2,557,734			2,660,850			2,700,004		
872,097			924,675			977,625		
100,000			100,000			100,000		
610,804			587,934			564,997		
$13,348,936			$12,818,006			$12,791,521		
	$114,152	1.97%		$117,092	2.25%		$117,416	2.34%
		4.00%			4.30%			4.31%

Income by Quarter *(In Thousands except per share amounts)*

	1984				1983			
	Fourth	*Third*	*Second*	*First*	*Fourth*	*Third*	*Second*	*First*
Interest Income								
Loans	$275,206	$271,550	$259,715	$239,117	$231,880	$223,095	$213,185	$214,122
Other	72,938	82,758	70,765	61,637	65,384	57,888	60,742	74,431
Total	348,144	354,308	330,480	300,754	297,264	280,983	273,927	288,553
Interest Expense								
Deposits	182,970	201,091	181,051	156,470	152,524	142,219	134,560	135,489
Other	59,664	53,410	51,728	42,876	41,363	40,857	45,151	54,785
Total	242,634	254,501	232,779	199,346	193,887	183,076	179,711	190,274
Net Interest Income	105,510	99,807	97,701	101,408	103,377	97,907	94,216	98,279
Provision for Loan Losses	22,984	15,566	31,651	23,802	19,411	15,499	13,977	17,500
Noninterest Income	59,519	53,919	71,100	58,731	51,945	50,700	49,745	46,440
Noninterest Expense	102,315	98,448	106,659	99,553	97,472	94,167	93,569	93,433
Income Before Income Taxes	39,730	39,712	30,491	36,784	38,439	38,941	36,415	33,786
Income Taxes	11,282	12,779	4,229	10,907	14,953	15,187	14,581	12,248
Net Income	$ 28,448	$ 26,933	$ 26,262	$ 25,877	$ 23,486	$ 23,754	$ 21,834	$ 21,538
Net Income Available to Common Shareholders	$ 26,212	$ 24,510	$ 24,050	$ 23,764	$ 21,335	$ 21,880	$ 19,700	$ 20,572
Per Common Share Data:								
Primary Earnings	$1.67	$1.57	$1.58	$1.58	$1.44	$1.51	$1.35	$1.40
Fully Diluted Earnings	1.40	1.32	1.32	1.32	1.20	1.24	1.12	1.16
Cash Dividends Declared	51	46	46	46	46	42½	42½	42½

Quarterly Earnings

During 1984 and 1983, earnings in each quarter increased from those recorded in the comparable year-earlier quarter by significant margins. Earnings in each of the quarters also exceeded those of the previous quarter, except for the fourth quarter of 1983, which trailed that year's third period by 1.1%.

For the fourth quarter of 1984, earnings increased by 21.1% to $28.4 million from the fourth quarter of 1983, with most major categories showing improvement. Fully diluted earnings per common share increased by 16.7% to $1.40. The gain in earnings came despite an 18.4% increase in the provision for loan losses, which management felt prudent, given economic uncertainties. Net charge-offs in the quarter were $8.6 million compared with $16.1 million in the fourth quarter of 1983. Net interest earnings on a taxable equivalent basis increased by 7.6% to $126.3 million as a result of expansion in the volume of loans, primarily domestic, and an increase in the value of noninterest-bearing sources of funds. The fourth quarter spread and net yield were 2.29% and 4.19% in 1984, respectively, versus 2.34% and 4.31% in 1983. Noninterest income was $59.5 million versus $51.9 million a year ago, an increase of 14.6%. The increase is attributable primarily to higher charges and fees and investment securities transactions. Noninterest expense in the quarter totaled $102.3 million, an increase of only 5.0% compared with the corresponding period in 1983.

Return on average assets for the quarter was .81% versus .73% in the year-earlier period, while return on average common shareholders' equity increased to 15.28% from 14.98% a year ago.

Noninterest Income

Total noninterest income, provided by a wide range of trading and fee-based services, increased 22.3% to $243.3 million in 1984, reflecting improvement in all categories. The most significant increase occurred in other operating income, and was attributable to a net gain of $24.0 million from the sale of branches. In 1984, the Company completed the sale of 36 branch offices in upstate New York; the former Capital Region (7 offices in and around Albany) was sold to Schenectady Trust Company, and the former Western, Southwestern and Southern Tier regions (29 offices in and around Buffalo, Olean and Endicott) were sold to Key Banks, Inc. Also included within this category are foreign exchange and trading account activity. Gains from foreign exchange trading amounted to $2.8 million, an increase of $1.7 million over 1983. Trading account profits totaled $1.2 million compared with $1.3 million in 1983.

Trust fees continue to represent a significant component of noninterest income, and include revenues from such diverse activities as personal trust services, investment advisory services, and various securities processing transactions. Due to the rapid growth within the latter area in the last several years, these activities require significant personnel and technological resources of the Company. Total trust fees increased $7.8 million, to $152.9 million, representing over 60% of total noninterest income.

The third major cause of the considerable increase from 1983 was investment securities gains, which totaled $693,000 compared to a loss of $5.5 million in 1983. The loss reported in 1983 was due to management's strategy of liquidating low-yielding earning assets.

Noninterest Expense and Income Taxes

Growth in noninterest expense was limited to 7.5% during 1984 and 5.4% during 1983, following an increase of 22.5% during 1982. In the fourth quarter of 1984, such expenses increased by only 5.0% compared with the corresponding period in 1983. Noninterest expense was $407.0 million in 1984 and $378.6 million the prior year. Primarily responsible for the substantially reduced rate of growth during the past two years were the continuing Company-wide cost consciousness program and, to a lesser extent, the sale of branches during the first half of the year. Lower growth in equipment requirements, as well as economies achieved in the operations sector of the Bank through expenditures made in previous years, was also responsible for the reduced growth rate of expenses over the past two years.

The Company's consolidated effective tax rate declined to 26.7% for the year from 38.6% a year earlier. The decline reflects the effects of a higher proportion of tax-exempt income, increased lease financing activities, and the impact of the capital gains tax rate applied to branch sales.

Liquidity

Maintaining liquidity requires having access to sufficient funds at reasonable rates to meet needs arising from asset growth and maturing liabilities. To assure a continued high degree of liquidity, the Company is as dependent upon its ability to renew existing liabilities and to sell additional ones as it is upon the ability of its assets to perform as contracted. Assurance of ready access to the capital and money markets requires that the Company maintain satisfactory earnings, high quality assets, a strong capital position and a solid reputation. The ability to manage interest rate risk is an interrelated and equally critical requirement for liquidity, as discussed more fully in the sections on Net Interest Earnings and Interest Sensitivity.

Indicative of the Company's liquidity has been its ability to raise funds both domestically and abroad at highly competitive rates. The introduction of money market deposit accounts has enhanced liquidity significantly by providing the Bank with a major new source of stable deposits at rates that compare favorably with those paid for purchased money. In recent years, the spread achieved by the Bank between federal funds sold and purchased has compared well with other money-center banks, as has the spread earned in its foreign offices between interest-bearing deposits in banks and foreign deposits. Additionally, rating agencies have assigned their highest rating to the Company's commercial paper, and in the debt markets during 1984 and in the equity markets during 1983, the Company was able to raise funds on terms competitive with the largest bank holding companies in the country. Also, throughout 1984 the Company was offered rates in the public markets for both debt and equity related issues that were competitive with rates achieved by many of the largest bank holding companies. With respect to the ability of the Company to obtain funds from its subsidiaries, refer to note 12 to the Consolidated Financial Statements on page 36. The regulations referred to in this note have not had a significant impact on the operations of the Company.

Interest Sensitivity

Constant growth in net interest earnings is dependent upon management's ability to maintain the appropriate degree of interest rate exposure. In order to maintain a consistent earnings flow in spite of fluctuations in the financial markets, the Company actively manages the maturity structure and rates of its interest-bearing assets and liabilities. The Company's sensitivity to interest rate movements is reflected in the interest-sensitive gap, which compares the repricing characteristics of assets and liabilities.

The following table reflects dollar assets and liabilities of the Company that will either be repriced in accordance with market rates or mature within the designated periods. Foreign currency assets and liabilities, which amount to $381.8 million, have been excluded since they are separately managed as yield curve structures differ from currency to currency.

Interest Sensitivity *(In Millions)*

	December 31, 1984				
	Within 1 Mo	*Within 2-3 Mos*	*Within 4-6 Mos*	*Greater Than 6 Mos*	*Total*
Interest-Earning Assets					
Foreign Offices	$1,840.2	$1,095.7	$567.9	$ 122.4	$ 3,626.2
Domestic Offices					
Gross Loans	4,742.4	249.5	182.8	2,333.4	7,508.1
Investment Securities	33.2	74.6	138.8	1,149.2	1,395.8
Trading Account Assets	66.6	–	–	–	66.6
Federal Funds Sold and Securities Purchased Under Resale Agreements	419.4	–	–	–	419.4
	7,101.8	1,419.8	889.5	3,605.0	$13,016.1
Interest-Bearing Liabilities					
Foreign Offices	2,183.0	590.5	416.2	207.2	$ 3,396.9
Domestic Offices					
Interest-Bearing Deposits	2,352.0	438.5	292.9	983.4	4,066.8
Federal Funds Purchased and Other Borrowed Funds	2,475.4	132.0	74.8	68.6	2,750.8
Long-Term Debt	–	–	–	169.2	169.2
	7,010.4	1,161.0	783.9	1,428.4	10,383.7
Noninterest-Bearing Sources of Funds	123.9	–	–	2,508.5	2,632.4
Total	7,134.3	1,161.0	783.9	3,936.9	$13,016.1
Interest-Sensitive Gap	$ (32.5)	$ 258.8	$105.6	$ (331.9)	
Cumulative Interest-Sensitive Gap	$ (32.5)	$ 226.3	$331.9	$ –	

The spring of 1984 was characterized by a rising rate environment, whereas monetary authorities substantially eased pressure on rates in the fall. The Company adjusted the interest-sensitive gap accordingly. As of year-end 1984, in the one-month period, which includes all loans with floating interest rates, the position was slightly liability-sensitive at $32.5 million. Compared to year-end 1983, when the one-month period showed a $741.4 asset-sensitive position, this year-to-year change represents a substantial decline in the size of the gap. Should rates start rising again, the Company could easily modify its position since, from a repricing perspective, most assets and liabilities are of a short-term nature.

In periods of changing interest rates, management sought a limited degree of mismatching in order to enhance earnings without absorbing undue risk. Additionally, steps were taken to limit the exposure of some accounts: consumer loans now include a variable-rate portion, and on the liability side, the Company issued variable-rate certificates of deposit.

Capital Resources

Management of the Company believes that its present capital base and its ability to access additional capital are adequate for both strong growth and contingencies. During 1984, the Company's average primary capital, as defined by the Federal Reserve Board to include equity accounts, the allowance for loan losses and certain qualifying debt, increased by 28.7% to $907.3 million from $705.1 million during 1983. For the third consecutive year, the growth in average capital exceeded the growth in average assets, with the result that the Company's primary capital ratio increased on an average basis to 6.78% from 5.55% in 1983 and 4.36% in 1982. At December 31, 1984, primary capital totaled $960.8 million, and the primary capital ratio was 6.34%. Subsequent to year end, on January 16, 1985, the Company called for redemption on February 15 of the balance of two series of convertible debentures, the outstandings of which at year-end totaled $85.4 million. The Company's common stock price is such that management believes virtually all debentures will be converted into common stock rather than redeemed. On a pro forma basis, giving effect to the conversion on December 31, 1984, primary capital would have been $1.0 billion and the primary capital ratio would have been 6.90%.

In January 1984, the primary capital position was augmented substantially with $75 million of floating rate substantially notes issued by the Company's overseas finance subsidiary in Europe on competitive terms. Also serving to augment capital were an additional $69.5 million in retained earnings, an increase in the allowance for loan losses of $49.3 million, $22.6 million from various employee and shareholder stock purchase programs, and $3.3 million from debenture conversions.

During 1983, the Company's capital position was increased substantially by a $100 million offering of adjustable rate cumulative preferred stock. Also serving to augment capital during 1983 were $58.7 million in retained earnings, a $17.9 million addition to the allowance for loan losses, $9.6 million from stock purchase programs, and $4.4 million from debenture conversions.

The Company's ability to access the public debt and equity markets continues to be strong. Although its plans during 1984 did not require additional funding, the Company was offered rates in these markets that were competitive with rates actually achieved by many of the country's largest bank holding companies. Giving effect to the conversion of its outstanding convertible debentures, the Company has substantial excess debt capacity, thus, affording it a high degree of flexibility in meeting growth requirements.

Loans

Three segments of the domestic loan portfolio which showed substantial growth on a year-end to year-end basis were commercial and industrial loans, construction and land development loans and loans for purchasing and carrying securities. Commercial and industrial loans increased by $805.8 million, or 29.2%, to $3.6 billion. Construction and land development loans increased by 28.3% to $898.7 million from $700.2 million at the end of the previous year. At year-end 1983, however, outstandings were somewhat depressed as a result of a continued, though improved, low level of business construction activity. Also, several major projects which the Bank had been financing on an interim basis were completed, and long term financing had been arranged. As measured by December 31, 1982 outstandings of $819.5 million, the portfolio's growth during the past two years has been 9.7%.

Loans for purchasing and carrying securities increased to $1.6 billion from $800.7 million at the end of the previous year. Average outstandings for the year, however, were $948.3 million, a level which more accurately reflects the growth of this portfolio over 1983 than does the year-end outstanding figure. These loans are of short maturity, predominately one day, and outstandings can and do fluctuate significantly within a short period of time.

Loans to individuals, which include installment, credit card and automobile loans, increased by 10.7% to $1.1 billion, continuing the strong growth of recent years. Growth in this portfolio has been achieved through extensive marketing efforts throughout the Bank's retail banking system. Growth from year-end 1983 to 1984 is understated by virtue of the sale of upstate branches during the first half of 1984.

Foreign loans, defined by the Securities and Exchange Commission (SEC) as loans to borrowers domiciled outside of the United States, increased by 3.9% to $1.6 billion. A major portion of these loans is to the foreign public sector and to commercial and industrial firms in Western Europe. For further discussion, refer to the section on Foreign Operations.

The Company again achieved record loan volume during 1984. Average loans, net of unearned income increased by 10.6% to $8.8 billion, following increases of 9.6% during 1983 and 21.2% during 1982. Loans outstanding on December 31, 1984 were $10.3 billion, an increase of 26.6% over the year earlier date. On an average basis, during the fourth quarter loans were $9.3 billion, compared with $8.1 billion in average outstandings during the fourth quarter of 1983, an increase of 14.9%.

The following table shows the Company's loan distribution at the end of each of the last five years.

Loan Portfolio by Type of Loans *(In Thousands)*

	December 31				
	1984	1983	1982	1981	1980
Domestic					
Real Estate Loans					
Construction and Land Development	$ 898,669	$ 700,246	$ 819,522	$ 864,398	$ 551,485
Secured by 1-4 Family Residential Properties	544,707	536,038	566,011	557,790	518,338
Secured by Multifamily Residential Properties	43,921	52,814	60,285	70,072	68,395
Other, Principally Commercial Mortgages	458,788	393,677	255,321	240,976	232,314
Loans to Domestic Banks	71,110	70,141	106,344	114,946	113,577
Loans for Purchasing or Carrying Securities	1,622,237	800,748	382,514	780,703	441,736
Commercial and Industrial Loans	3,568,150	2,762,358	2,748,050	2,117,881	1,873,514
Loans to Individuals	1,066,267	963,138	808,650	767,728	661,574
Lease Financings	265,167	128,524	77,424	66,978	43,516
All Other	407,684	368,959	483,926	418,310	422,860
Total Domestic	8,946,700	6,776,643	6,307,847	5,999,782	4,927,309
Foreign					
Governments and Official Institutions	571,178	614,092	432,239	394,897	299,505
Banks and Other Financial Institutions	386,652	383,454	415,539	304,036	342,174
Commercial and Industrial	583,801	485,560	496,942	406,428	254,514
All Other	11,448	11,213	7,808	4,633	3,876
Total Foreign	1,553,079	1,494,319	1,352,528	1,109,994	900,069
	10,499,779	8,270,962	7,660,375	7,109,776	5,827,378
Less: Unearned Income	196,450	130,475	107,503	91,661	82,395
Allowance for Loan Losses	131,250	82,000	64,100	44,300	36,800
Net Loans	$10,172,079	$8,058,487	$7,488,772	$6,973,815	$5,708,183

The following table shows the maturity structure of the Company's loan portfolio at December 31, 1984.

Loan Maturity *(In Thousands)*

	December 31, 1984			
	1 Year or Less	*Over 1 Year Through 5 Years*	*Over 5 Years*	*Total*
Domestic				
Real Estate, Excluding Loans Secured by 1-4 Family Residential Properties	$ 617,691	$ 541,967	$ 241,720	$1,401,378
Commercial and Industrial Loans	2,163,600	956,036	448,514	3,568,150
Other, Excluding Loans to Individuals and Those Secured by 1-4 Family Residential Properties	1,941,948	143,174	281,076	2,366,198
	4,723,239	1,641,177	971,310	7,335,726
Foreign	969,121	319,934	264,024	1,553,079
Total	$5,692,360	$1,961,111	$1,235,334	$8,888,805
Loans with				
Predetermined Interest Rates	$2,091,466	$ 510,466	$ 565,751	$3,167,683
Floating Interest Rates	3,600,894	1,450,645	669,583	5,721,122
Total	$5,692,360	$1,961,111	$1,235,334	$8,888,805

Provision and Allowance for Loan Losses

Evaluating credit risk and minimizing exposure to loss are two critical functions of bank management. The risk of loss is inherent in lending, and implementation of effective credit review procedures to minimize loss, consistent with the need to meet the legitimate credit needs of customers, is essential to sound banking practice. In recent years, the credit evaluation process has become an even more integral determinant of profitability for the banking industry as increased competition from both domestic and foreign sources has resulted in narrowed spreads and consequent impairment of the industry's ability to absorb credit losses out of net interest income. Concurrent with these developments, inflation, high long-term interest rates, lower productivity, weakened capital markets and sharply reduced rates of savings and capital formation have resulted in substantially increased reliance on the banking industry by increasingly less liquid and more highly leveraged borrowers in both the private sector and foreign public sector.

The risk profile of the Company's loan portfolio is believed by management to be relatively low, and its lending activities are heavily oriented toward domestic borrowers. Domestically, management controls loss exposure by diversifying its portfolio among various categories of borrowers. In its commercial and industrial loan portfolio, borrowers represented include those engaged in heavy and light manufacturing, transportation, communication and other service-oriented businesses, without significant concentration relative to one another.

Although construction and land development portfolio outstandings were up substantially from a depressed year-end 1983 level, outstandings were up only 9.7% from year-end 1982, as previously discussed. In order to control exposure in this segment of the portfolio, management restricts lending activity generally to first mortgage loans in projects being developed by groups which have demonstrated a strong performance record.

The Company's large portfolio of loans for purchasing or carrying securities is comprised largely of overnight loans which are generally fully secured with appropriate margins by marketable securities. The country's largest and most established brokerage houses are the major borrowers.

Among foreign borrowers, a significant portion of the Company's exposure is limited to major financial institutions, central banks, governments and their agencies, and corporate and commercial loans that are either well secured and/or well guaranteed. For further discussion, refer to the section on Foreign Operations.

The Company's management establishes and continually reviews lending policies to determine the overall risk profile and quality of the loan portfolio. Factors in the evaluation process are considered in the context of current and expected economic conditions and their impact on the existing loan portfolio. Analytical reviews of loan loss experience, problem loans, and loan concentrations in terms of industry, geography and size are integral to this review.

To provide for potential loan losses, the Company maintains an allowance for loan losses, against which loan losses are charged and loan recoveries are credited. Additions to the allowance are made through provisions charged against income.

During the past five years, the provision charged to income greatly exceeded net charge-offs. During 1984, the provision charged to income was $94.0 million, while net charge-offs were $44.8 million. In the prior year, the provision was $66.4 million, while net charge-offs were $48.5 million, and in 1982 the provision was $49.1 million, while net charge-offs were $29.3 million. Gross charge-offs were $52.4 million in 1984, $54.6 million in 1983, and $32.9 million in 1982, while recoveries were $7.7 million, $6.1 million, and $3.6 million, respectively.

Net charge-offs as a percent of average loans outstanding were .51% during 1984 compared with .61% during 1983 and .40% during 1982. Although net charge-offs as a percent of average loans declined during 1984, the generally higher levels of the past two years relative to 1982, 1981 and 1980 reflect generally weak economic conditions prevailing prior to last year both domestically and abroad, as well as a conservative charge-off policy. Since the provision charged to income greatly exceeded net charge-offs, the allowance for loan losses at year end increased to $131.3 million from $82.0 million in 1983 and $64.1 million in 1982. As a result, the allowance as a percentage of loans outstanding at year-end increased to 1.27% compared with 1.01% in 1983 and .85% in 1982. During the past several years the substantial increases in the provision charged to income and the allowance for loan losses reflected a larger loan portfolio, but also a management decision to assume a prudent and cautious approach to economic and business uncertainties, particularly with respect to foreign loans. Although loan charge-offs may vary widely due to general economic conditions and specific developments affecting particular industries and borrowers, and therefore cannot be predicted accurately, total net charge-offs as a percentage of average loans outstanding in 1985 are expected to approximate the average for the past year.

The following table sets forth details by major loan category with respect to the Company's allowance for loan losses and loans, net of unearned income, for each of the last five years.

Analysis of the Allowance for Loan Losses *(In Thousands)*

	1984	1983	1982	1981	1980
Loans Outstanding, December 31	$10,303,329	$8,140,487	$7,552,872	$7,018,115	$5,744,983
Average Loans Outstanding	$ 8,782,494	$7,941,010	$7,242,641	$5,974,910	$4,805,104
Allowance for Loan Losses					
Balance, January 1					
Domestic	$ 67,000	$ 61,100	$ 42,300	$ 35,400	$ 24,800
Foreign	15,000	3,000	2,000	1,400	1,300
Total Balance, January 1	82,000	64,100	44,300	36,800	26,100
Loans Charged Off					
Domestic					
Commercial and Industrial Loans	(35,577)	(33,125)	(14,016)	(9,362)	(6,286)
Loans to Individuals	(13,227)	(13,278)	(8,349)	(7,689)	(5,804)
Other	(811)	(750)	(7,529)	(1,091)	(270)
Foreign	(2,815)	(7,450)	(3,000)	—	—
Total	(52,430)	(54,603)	(32,894)	(18,142)	(12,360)
Recoveries					
Domestic					
Commercial and Industrial Loans	3,564	3,564	1,399	4,028	909
Loans to Individuals	3,691	2,467	2,144	2,009	1,646
Other	404	85	39	81	31
Foreign	18	—	—	—	—
Total	7,677	6,116	3,582	6,118	2,586
Net Charge-offs	(44,753)	(48,487)	(29,312)	(12,024)	(9,774)
Balance Relating to Acquired Bank	—	—	—	—	3,704
Provision Charged to Income					
Domestic	76,206	46,937	45,112	18,924	16,670
Foreign	17,797	19,450	4,000	600	100
Total	94,003	66,387	49,112	19,524	16,770
Balance, December 31					
Domestic	101,250	67,000	61,100	42,300	35,400
Foreign	30,000	15,000	3,000	2,000	1,400
Total Balance, December 31	$ 131,250	$ 82,000	$ 64,100	$ 44,300	$ 36,800
Ratios					
Net Charge-offs to Average Loans Outstanding	.51%	.61%	.40%	.20%	.20%
Net Charge-offs to Total Allowance	34.10%	59.13%	45.73%	27.14%	26.56%
Allowance to Year-end Loans Outstanding	1.27%	1.01%	.85%	.63%	.64%

Risk Elements

There are various risk elements, as defined by the SEC, inherent in bank lending activities. Such risk elements include nonaccrual, reduced rate and past due loans, potential problem loans, foreign outstandings and loan concentrations. All of these elements are described in detail below except for loan concentrations, since a discussion is included in the sections on Loans and Provision and Allowance for Loan Losses.

Nonaccrual, Reduced Rate, Past Due and Potential Problem Loans

The Company's policy is to discontinue the accrual of interest on loans (other than installment loans and 1-4 family residential mortgages) on which principal or interest is past due and remains unpaid for 90 days or more and which are not fully collateralized, or whenever there is reasonable doubt that interest and/or principal will be received. Such loans are classified as nonaccrual by the Company. Loans which reach nonaccrual status are not restored to accruing status until all delinquent principal and/or interest has been brought current or the loan becomes both fully collateralized and in the process of collection. This classification, however, does not necessarily indicate that the principal of the loan is uncollectable but does generally warrant a reevaluation of collectability. When a loan is placed on nonaccrual status, generally any unpaid interest previously accrued is reversed against current income. Thereafter, interest may be credited to income on a cash basis or applied to principal, depending upon management's judgment of the extent of the credit risk involved.

If a partial charge-off is taken on any loan, the remaining balance automatically is placed on a nonaccrual status and is accounted for on a cost-recovery basis, that is, any interest and principal payments received are applied first to the reduction of net book principal, then to the recovery of principal previously charged off, and finally to interest income.

The Company does not place installment loans (primarily representing consumer loans) on a nonaccrual basis, but instead charges off all such loans when they become 180 days past due.

The Company also modifies the terms of certain loans by reducing the interest rates and extending the maturities to improve ultimate collections. Such loans are classified as reduced rate by the Company.

Information with respect to the Company's nonaccrual and reduced rate loans is presented in the following tables.

Nonaccrual and Reduced Rate Loans and Real Estate Acquired in Satisfaction of Loans *(In Thousands)*

	December 31,				
	1984	1983	1982	1981	1980
Nonaccrual					
Domestic	$229,416	$192,593	$153,834	$125,549	$80,251
Foreign	45,781	29,066	15,841	—	—
	275,197	221,659	169,675	125,549	80,251
Reduced Rate					
Domestic	16,798	19,397	41,368	41,114	2,169
Foreign	—	—	2,658	—	—
	16,798	19,397	44,026	41,114	2,169
	291,995	241,056	213,701	166,663	82,420
Real Estate Acquired in Satisfaction of Loans	22,208	20,696	45,786	6,064	5,778
	$314,203	$261,752	$259,487	$172,727	$88,198

	Year Ended December 31, 1984			
	Nonaccrual		Reduced Rate	
	Domestic	*Foreign*	*Domestic*	*Foreign*
Interest Income That Would Have Been Recorded Under Original Terms	$30,630	$5,635	$1,422	$ —
Interest Income Recorded During the Period	5,576	2,141	903	—

Nonaccrual and reduced rate loans totaled $292.0 million at December 31, 1984 versus $241.1 million at December 31, 1983, an increase of $50.9 million. As a percent of total loans outstanding, nonaccrual and reduced rate loans at December 31 decreased to 2.8% in 1984 from 3.0% in 1983. The current year's increase in nonaccrual and reduced rate loans was primarily attributable to two real estate construction projects and a loan to a diversified financial concern.

Of the total year-end 1984 nonaccrual and reduced rate loans, over 40% related to real estate construction projects, 11.1% to the oil and gas industry and 15.7% to borrowers in foreign countries. The principal of the real estate loans is considered to be adequately secured.

Foreign nonaccrual loans totaled $45.8 million and included $12.0 million in Argentina, $12.1 million in Poland and $13.7 million in Peru.

Past due loans that are fully collateralized as to principal and interest are not placed on a nonaccrual basis according to the Company's policy. The principal amount of all such loans past due 90 days or more was primarily domestically derived and amounted to $166.1 million, $152.1 million, $146.5 million and $87.2 million at December 31, 1984, 1983, 1982 and 1981, respectively.

At December 31, 1984, there were two loans totaling $52.5 million considered potential problem loans as defined by the SEC.

Foreign Outstandings

The Company's cross-border outstandings (loans, interest-bearing deposits in banks and similiar assets) to borrowers in foreign countries where such outstandings exceeded .75% of total assets at December 31, 1984 and 1983 were as follows:

Country	*Governments and Official Institutions**	*Banks and Other Financial Institutions*	*Commercial and Industrial Companies*	*Other*	*Total*
			(In Millions)		
			December 31, 1984		
United Kingdom	$ 1.8	$191.4	$344.8	$ —	$538.0
Brazil	76.2	191.0	3.8	—	271.0
Mexico	118.2	102.1	44.5	—	264.8
Italy	29.5	195.7	6.1	—	231.3
Canada	17.3	125.0	68.0	—	210.3
Japan	—	180.6	.3	—	180.9
Philippines	64.8	40.4	11.0	—	116.2
			December 31, 1983		
United Kingdom	$ 6.3	$167.5	$366.6	$ —	$540.4
Mexico	142.4	97.0	26.2	—	265.6
Brazil	17.9	230.8	—	—	248.7
Italy	50.8	191.0	3.4	—	245.2
France	—	163.7	6.9	—	170.6
Spain	17.5	137.5	8.0	—	163.0
Canada	24.6	95.3	25.4	4.9	150.2
Japan	—	147.1	.2	—	147.3
Philippines	63.9	38.4	12.4	—	114.7
Korea	56.6	49.9	—	—	106.5

*Includes loans to commercial and industrial companies owned 50% or more by the governments of Brazil and Mexico.

Since mid-1982, certain countries have encountered significant and continuing difficulties in servicing their foreign-held debt. Brought on by severe recessionary conditions throughout the world economies, this situation was, in certain instances, exacerbated by political events. The resultant effects include a shortage of foreign exchange availability and a greatly restricted access to the capital markets.

In response, the International Monetary Fund (IMF), in collaboration with a number of Western governments and the commercial banking community, has tailored a series of financial assistance packages for these countries. These programs include the setting of economic targets which, through their achievement, would stabilize and reorder economic conditions, allowing for an orderly repayment of existing debt and revival of the capital markets for these countries. The Bank has participated in several of these programs, including those commented on below, and expects to participate further in others.

Brazil

During 1984, Brazil made major strides in resolving its severe temporary liquidity problems encountered the previous year with the successful completion in February of the IMF-backed Phase II finance package. The package included a $6.5 billion, nine-year term loan (Bank share-$26.3 million), which was fully drawn down by December as Brazil continued to meet all IMF requirements for disbursement. Other aspects of the Phase II package were included in the refinancing of 1984 maturities of medium-term loans for nine years (Bank share-$.9 million) as well as commitments to maintain until February 1985 trade finance (Bank share-$179.8 million) and interbank deposit (Bank share-$21.0 million) facilities which are equal to outstandings as of June 30, 1983. The Bank committed to its participation.

In January 1985, negotiations between the banks and the Brazilians toward a Phase III, 1985 financing package were in their final stages. Due to Brazil's improved financial condition in 1984, especially in the area of its external accounts, where a record trade surplus was achieved, Brazil announced it would not require new money as part of Phase III. A proposal for multi-year refinancing of maturities of medium-term debt was under discussion. During 1984 interest income of $10.5 million was accrued on Brazilian loans and $8.0 million was collected. At year-end 1984, Brazil was meeting all interest payments on a timely basis, and it is anticipated that under Phase III all interest payments will continue to be made when due. No Brazilian debt was included in reported amounts of nonaccrual, reduced rate and past due loans.

Mexico

On September 8, 1984, the Mexican authorities proposed a multi-year restructuring of public sector debt. The 13-member Bank Advisory Group supports the proposal and has urged all lenders to accept its terms. Under the proposal, which is projected for signing during the first quarter of 1985, the Bank would extend $87.0 million of maturities falling due between 1983 and 1990 for fourteen years, with a one-year grace period. Additionally, $42.5 million of 1987 through 1990 maturities under existing Restructuring Agreements will be extended by eleven years (the 1987 maturities) and by ten years (the 1988-90 maturities). No grace period is contemplated. All of these loans will carry the full guarantee of the United Mexican States. The final category in the restructuring package affects the $5.0 billion 1983 new-money facility in which the Bank participates with $17.5 million. The Mexican government proposed making a partial prepayment of the facility which would reduce the Bank's outstandings to $14.0 million. All restructured debt will have new pricing as follows:

Financing	*Amount (millions)*	*Weighted Average Rate*	
		Before Restructuring	*After Restructuring*
I	$14.0(1)	2.125% over prime	1.125% over prime
II	41.6	0.78% over prime	Initially 0.875% over LIBOR or a comparable domestic reference rate (2)(3)
	37.6	1.16% over LIBOR	
	7.8	0.625% over SDR (Special Drawing Rights)	
III	42.5	1.75% over prime	Initially 0.875% over LIBOR or a comparable domestic reference rate (2)(3)

(1) Does not include $3.5 million due to be prepaid in 1985.
(2) Rates will be .875% as shown for the period 1985-1986, 1.125% for the period 1987-1991 and 1.250% for the period 1992-1998.
(3) At December 31, 1984, the prime rate and the SDR rate exceeded LIBOR by approximately 2.06% and 0.69%, respectively.

During the negotiations, the Mexican government maintained that it would seek new money only on a voluntary market basis. Additional public sector outstandings as of December 31, 1984 not included in the above table were $108.0 million, of which $88.5 million was short-term and $19.5 million was medium-term. Also included in the $108.0 million are new cross-border advances to the United Mexican States of $10.5 million. No principal was repaid on public sector debt during this time. Private sector Mexican indebtedness amounted to $13.3 million at December 31, 1984. During 1984, interest income of $28.2 million was accrued on Mexican loans and $24.7 million was collected. At December 31, 1984, all Mexican indebtedness was current as to payment of interest. No Mexican debt was included in reported amounts of nonaccrual, reduced rate and past due loans at December 31, 1984.

Philippines

In 1983, political turbulence, an unfavorable economic climate and severe capital flight caused an acute liquidity crisis. In light of the reduced level of foreign exchange reserves, on October 17, 1983 the Central Bank of the Philippines requested and was granted a 90-day moratorium on principal payments of foreign-held debt maturing during the ensuing 90-day period. Subsequent additional 90-day extensions were requested and granted, the latest on January 8, 1985. The moratoriums are intended to provide a stable climate for discussions with the IMF and a representative group of international banks composing the Bank Advisory Committee. On November 19, 1984, the Philippines presented a rescheduling program. The IMF supported the rescheduling program, and the Bank Advisory Committee recommended approval by the international banking community. Approval by the international banks was received, and an agreement is expected to be signed in early 1985. The restructuring program calls for new money of $925 million (Bank share-$8.6 million). The restructuring package also includes a revolving short-term trade facility available for two years totaling $3.1 billion (Bank share-$54.1 million). In addition, there will be a restructuring of public and private, nontrade, short-term and medium-term debt with maturities falling due between October 17, 1983 and December 31, 1986. All public sector debt falling due during this period will be extended for ten years, with a five-year grace period at 1.625% over LIBOR (Bank share-$15.3 million). The private financial sector short-term debt maturing during the same period either will be extended for four years or the lender has the option to convert such private debt into a ten-year maturity as a public sector obligation (Bank share-$4.0 million). The private corporate industrial sector debt will be negotiated and restructured separately (the Bank has $12.5 million under this category). Other Bank outstandings not restructured are $28.3 million. No Philippine debt was included in reported amounts of nonaccrual, reduced rate and past due loans at December 31, 1984.

Special Reserves

Acting under the International Lending Supervision Act, the Federal Reserve Board has mandated the establishment of special reserves against loans to borrowers in five countries, or in lieu of special reserves a charge-off of an equivalent amount. The Bank had loans to borrowers in two of those countries and had, prior to year-end 1984, charged off loans to those countries in excess of the required reserves. These charge-offs amounted to $1.8 million in 1984 and $4.5 million in 1983.

Securities

On December 31, 1984, the total investment portfolio amounted to $1.5 billion, an increase of 15.5% from the balance of $1.3 billion at year-end 1983. The largest component of the portfolio remained obligations of states and political subdivisions, representing 52.9% of the total and increasing by $136.2 million from the prior year. This category will continue to represent the principal source of tax-exempt income. Holdings of U.S. Government and Government agency obligations were up $29.0 million, to $602.5 million, or 40.7% of the portfolio. The following table sets forth the carrying amount of investment securities at the dates indicated.

Investment Portfolio—Book Values *(In Thousands)*	*December 31* 1984	1983	1982
U.S. Government Obligations	$ 548,624	$ 520,350	$ 327,419
U.S. Government Agency Obligations	53,850	53,135	63,405
Obligations of States and Political Subdivisions	783,281	647,049	586,560
Other Securities	95,158	61,227	44,592
	$1,480,913	$1,281,761	$1,021,976

The following table shows the maturity distributions by book value and yield (not on a taxable equivalent basis) of the Company's investment portfolio at December 31, 1984.

Investment Portfolio—Maturity Distribution *(In Thousands)*

	December 31, 1984										
	U.S. Government		*U.S. Government Agency*		*States and Political Subdivisions*		*Other Bonds, Notes and Debentures*		*Stocks and Warrants*		
	Amount	*Yield*	*Amount*	*Yield*	*Amount*	*Yield*	*Amount*	*Yield*	*Amount*	*Yield*	*Total*
One Year or Less	$131,837	11.16%	$ 6,997	7.35%	$246,705	6.43%	$ 4,547	9.52%	$ —	— %	$ 390,086
Over 1 Through 5 Years	351,481	10.78	38,018	7.57	40,272	7.38	7,003	10.58	—	—	436,774
Over 5 Through 10 Years	60,063	10.97	—	—	31,589	5.80	417	11.00	—	—	92,069
Over 10 Years	5,243	11.29	8,833	13.03	464,715	9.01	—	—	—	—	478,791
No Maturity	—	—	—	—	—	—	—	—	83,191	5.86	83,191
	$548,624	10.90%	$53,850	8.44%	$783,281	7.98%	$11,967	10.19%	$83,191	5.86%	$1,480,913

Deposits

Deposits continue to represent the Company's principal source of funds and consist of demand deposits (noninterest-bearing), money market deposits, savings deposits and various forms of time deposits. At year-end 1984, total deposits amounted to $10.5 billion, an increase of $968.3 million over 1983.

Average deposits in 1984 were up $505.0 million to $9.7 billion, an increase of 5.5%. Much of this rise is attributable to the popularity of the money market deposit accounts, introduced in December 1982. Average demand deposits declined by $225.8 million in 1984, a reflection of the variety of attractive interest-bearing products available to the consumer. The sale of the upstate branches also contributed to the reduction in demand deposits.

The following table summarizes average deposits and rates paid thereon for the periods indicated.

Average Deposits and Average Rates *(In Thousands)*

	1984		1983		1982	
	Amount	*Average Rate*	*Amount*	*Average Rate*	*Amount*	*Average Rate*
Domestic Offices						
Noninterest-Bearing Demand	$2,512,123		$2,739,673		$2,486,548	
Money Market Deposits	1,613,550	9.24%	1,223,663	8.37%	5,332	9.79%
Savings	752,067	5.08	878,958	5.08	977,753	5.11
Time	1,827,347	11.25	1,519,752	9.44	2,317,784	12.37
	6,705,087		6,362,046		5,787,417	
Foreign Offices						
Banks in Foreign Countries (Including Foreign Branches of Other U.S. Banks)	1,180,808		1,134,568		1,794,583	
Governments and Official Institutions	202,418		145,689		273,431	
Other Demand Deposits	2,463		3,619		3,099	
Other Time and Savings	1,588,350		1,528,215		1,066,928	
	2,974,039	11.05%	2,812,091	9.75%	3,138,041	13.41%
	$9,679,126		$9,174,137		$8,925,458	

The aggregate amount of deposits by foreign depositors in domestic offices was $160.4 million, $266.7 million and $294.6 million at December 31, 1984, 1983 and 1982, respectively.

The following table shows the maturity breakdown of time deposits of $100,000 or greater at year-end 1984.

Maturity of Domestic Time Deposits of $100,000 or Greater *(In Thousands)*

	December 31, 1984		
	Certificates of Deposit	*Time Deposits–Other*	*Total*
3 Months or Less	$433,442	$625,475	$1,058,917
Over 3 Through 6 Months	39,004	43,654	82,658
Over 6 Through 12 Months	6,825	111,127	117,952
Over 12 Months	56,910	7,733	64,643
	$536,181	$787,989	$1,324,170

The majority of foreign deposits are time deposits in denominations of $100,000 or greater.

Short-Term Borrowings

Federal funds purchased and securities sold under repurchase agreements generally represent overnight borrowing transactions. Commercial paper is issued by the Parent Company generally in amounts of not less than $100,000, with maturities of 180 days or less. Other short-term borrowings principally consist of extended federal funds purchased and U.S. Treasury note balances which are payable on demand. Information related to these borrowings in 1984, 1983 and 1982 is presented in the table below. The rates shown have been computed by dividing total interest expense by the average amount borrowed.

Short-Term Borrowings *(In Thousands)*

	1984		1983		1982	
	Amount	*Average Rate*	*Amount*	*Average Rate*	*Amount*	*Average Rate*
Federal Funds Purchased and Securities Sold Under Repurchase Agreements						
At December 31	$2,166,434	8.12%	$ 841,120	9.59%	$1,883,674	10.95%
Average during year	1,262,193	9.91	1,316,904	8.87	1,3[illegible]3,430	12.14
Maximum month-end balance during year	2,215,333	9.84	2,744,346	8.35	1,883,674	10.95
Commercial Paper						
At December 31	149,965	8.56	44,997	9.70	98,526	9.23
Average during year	80,105	10.76	50,293	9.26	27,145	11.38
Maximum month-end balance during year	187,692	11.45	85,129	9.03	98,526	9.23
Other Short-Term Borrowings						
At December 31	434,750	9.35	601,195	9.60	240,495	10.30
Average during year	523,412	10.49	505,671	9.70	462,307	14.26
Maximum month-end balance during year	767,786	10.96	621,789	9.60	727,290	14.23

Foreign Operations

Interest-bearing deposits in banks decreased 10.2% to $992.3 million, on an average basis, as more attractive investment alternatives became available. At December 31, 1984, deposits were primarily with major foreign banks in Europe and the Western Hemisphere.

Total average foreign loans increased 11.7% to $1.6 billion in 1984; of this total, 37.2% were loans to governments and official institutions, 36.4% were commercial and industrial loans and 26.1% were loans to financial institutions.

Average foreign deposits remained virtually unchanged at $1.9 billion in 1984. Deposits of banks in foreign countries, which include foreign branches of other U.S. banks, accounted for 49.5% of this total, other time deposits accounted for 28.8%, and other demand deposits accounted for 13.2%.

Foreign other liabilities increased 60.0% to $205.5 million on an average basis. This increase is primarily attributable to the issuance of floating rate notes totaling $75.0 million by a wholly owned foreign subsidiary.

Average Foreign Assets and Liabilities *(In Millions)*

	1984	1983	1982
Average Foreign Assets			
Interest Bearing Deposits in Banks	$ 992.5	$1,104.6	$1,490.0
Loans			
Financial Institutions	411.3	410.8	318.7
Governments and Official Institutions	586.8	488.0	375.0
Commercial and Industrial	573.9	508.5	477.3
Other	5.3	4.4	.6
Total Loans	1,576.9	1,411.7	1,171.6
Due from Customers on Acceptances	527.8	565.5	536.2
Other Assets	90.4	85.1	48.0
Total Assets	$3,187.4	$3,166.9	$3,245.8
As Percentage of Consolidated Average Assets	23.8%	24.9%	26.5%
Average Foreign Liabilities			
Deposits			
Banks in Foreign Countries (Including Foreign Branches of Other U.S. Banks)	$ 919.9	$ 951.0	$1,566.7
Governments and Official Institutions	190.1	162.2	220.8
Other Demand and Time	780.2	737.8	412.4
Total Deposits	1,890.2	1,851.0	2,199.9
Acceptances Outstanding	527.8	565.5	536.4
Other Liabilities	205.5	128.4	43.1
Total Liabilities	$2,623.5	$2,544.9	$2,779.4
As Percentage of Consolidated Average Liabilities	20.9%	21.1%	23.6%

Geographic Distribution of Average Foreign Assets

Although the Company's foreign assets are widely distributed, its international operations are primarily centered in European countries, which accounted for 46.0% of the foreign assets in 1984, followed by Western Hemisphere countries, which accounted for 32.3% of the foreign assets. The United Kingdom is the only country accounting for more than 10% of the Company's total foreign assets in 1984. Based upon the economic categories used by the International Bank for Reconstruction and Development/The World Bank, 51.0% of foreign asset exposure at year-end 1984 was to industrial countries, 42.7% to countries classified in the upper and intermediate middle income groups, and only 4.2% to lower middle and low income countries.

Average Foreign Assets by Geographical Region *(In Millions)*

	Average Total Assets for the Twelve Months Ended December 31, 1984	
	Amount	*Percentage*
Europe	$1,467.1	46.0%
Middle East and Africa	75.9	2.4
Asia and Pacific	613.7	19.3
Western Hemisphere	1,030.7	32.3
Total	$3,187.4	100.0%

Year-End Total Foreign Assets by Economic Category Used by the International Bank for Reconstruction and Development/The World Bank *(In Millions)*

	Total Assets as of December 31, 1984	
	Amount	*Percentage*
Industrial Market Economies	$1,548.6	51.0%
Nonmarket Industrial Economies	31.0	1.0
Capital Surplus Oil Exporters	34.8	1.1
Upper Middle Income Countries (Per capita income—$3,000 and over)	195.5	6.4
Intermediate Middle Income Countries (Per capita income—$700 to $2,999)	1,102.2	36.3
Lower Middle Income Countries (Per capita income—$500 to $699)	115.1	3.8
Low Income Countries (Per capita income—Less than $500)	10.6	.4
	$3,037.8	100.0%

Reporting the Effects of Inflation

The rate of inflation in recent years has led to concern that historical financial data do not adequately report results in this environment. Present financial statements include the actual dollars received or expended without regard to the variability in the purchasing power of the dollar. The financial data presented in the accompanying table have been prepared in accordance with the requirements of Financial Accounting Standards Board Statement No. 33, as amended, which attempts to measure the impact of inflation on a business enterprise. The information is intended to supplement, but not replace, the basic financial statements and other traditional financial data presented throughout this report.

The following table presents a five-year comparison of selected financial data in historical dollars and average 1984 constant dollars. The historical cost financial data are based on amounts reflected in the Company's financial statements presented elsewhere in the report. The amounts adjusted for general inflation are restated on a constant dollar basis, which is expressed in average 1984 dollars and reflects adjustments that have occurred in the purchasing power of the dollar as measured by the Consumer Price Index for All Urban Consumers published by the Bureau of Labor Statistics. The constant dollar income (except net interest income) and net asset amounts have been adjusted to reflect premises and equipment as well as depreciation expense on a constant dollar basis.

It should be noted that distortions may be inherent in applying the Consumer Price Index to foreign operations and all assets, liabilities, income and expense amounts of the Company. In order to assess the Company's ability to cope with inflation, reference is also made to disclosures contained elsewhere in this report regarding interest rate sensitivity and asset and liability maturities.

Five-Year Comparison of Selected Supplementary Financial Data
Adjusted for Effects of Changing Prices (1)
(In Average 1984 Dollars except "As Reported" amounts)

	Year Ended December 31,				
	1984	1983	1982	1981	1980
	(In Millions except per share amounts)				
Net Interest Income					
As Reported	$404.4	$393.8	$357.9	$277.5	$223.3
Adjusted for General Inflation	404.4	410.5	385.1	316.9	281.5
Income Before Extraordinary Charge					
As Reported	107.5	90.6	73.2	58.2	48.8
Adjusted for General Inflation (2)	102.2	99.0	73.2	61.1	56.2
Net Income					
As Reported	107.5	90.6	73.2	58.2	42.6
Adjusted for General Inflation (2)	102.2	99.0	73.2	61.1	48.4
Primary Per Common Share Information:					
Income Before Extraordinary Charge					
As Reported	6.40	5.70	5.09	4.25	3.74
Adjusted for General Inflation (2)	6.06	5.57	5.09	4.46	4.32
Net Income					
As Reported	6.40	5.70	5.09	4.25	3.27
Adjusted for General Inflation (2)	6.06	5.57	5.09	4.46	3.72
Purchasing Power Loss on Net Monetary Items (3)	22.4	17.7	13.8	29.3	43.1
Net Assets at Year-End					
As Reported	760.3	664.0	505.2	444.2	394.3
Adjusted for General Inflation	801.8	731.6	599.5	558.6	536.5
Cash Dividends Declared Per Common Share					
As Reported	1.890	1.735	1.625	1.525	1.395
Adjusted for General Inflation	1.890	1.809	1.749	1.742	1.758
Market Price Per Common Share at Year-End					
As Reported	35.50	32.75	24.88	21.44	17.38
Adjusted for General Inflation	35.00	33.57	26.47	23.65	20.92
Average Consumer Price Index	311.1	298.4	289.1	272.4	246.8

(1) The data presented are based upon a historical cost/constant dollar method of accounting. Because current cost amounts do not differ materially from historical cost/constant dollar amounts, current cost data have not been included.
(2) The adjustment for depreciation and amortization expense was $5.3 million in 1984.
(3) All assets and liabilities except premises and equipment are considered monetary for purposes of these calculations. As the Company's monetary assets are greater than its monetary liabilities, a net increase in the Consumer Price Index would always result in a decline in the purchasing power of net assets.
Notes: The computations of the supplementary information exclude any adjustments to the income tax expense included in the historical cost financial statements.

Financial Ratios

The Company's significant ratios for the five years ended December 31, 1984 are as follows:

	1984	1983	1982	1981	1980
Earnings Ratios					
Net Income Before Extraordinary Charge as a % of:					
Average Earning Assets	.94%	.85%	.72%	.63%	.60%
Average Total Assets	.80	.71	.60	.52	.48
Average Common Shareholders' Equity	17.14	16.75	15.30	13.91	12.68
Average Shareholders' Equity	14.78	14.47	15.30	13.91	12.68
Net Income Before Extraordinary Charge Available to Common Shareholders as a % of Average Common Shareholders' Equity	15.71	15.43	15.30	13.91	12.68
Capital Ratios					
Average Loans as a Multiple of Average Shareholders' Equity	12.07 ×	12.68 ×	15.13 ×	14.29 ×	12.49 ×
Average Shareholders' Equity as a % of Average Total Assets	5.44%	4.93%	3.91%	3.72%	3.82%
Average Primary Capital as a % of Average Total Assets	6.78	5.55	4.36	4.10	4.13
Primary Capital as a % of Year-End Total Assets	6.34	5.83	4.47	4.26	4.14
Other					
Allowance for Loan Losses as a % of Loans	1.27%	1.01%	.85%	.63%	.64%
Average Annual Dividend Yield	6.2	6.0	7.6	8.0	8.3
Dividend Payout Ratio	29.53	30.44	31.93	35.88	42.66
Average Annual Price/Earnings Ratio	4.8 ×	5.1 ×	4.1 ×	4.5 ×	4.3 ×

Market Price Information

The high and low sale prices of the Company's common stock on the New York Stock Exchange for each quarter of 1984 and 1983 were:

1984	1st	2nd	3rd	4th
High	34 1/4	31 3/8	32 1/8	36 3/4
Low	28	26 7/8	26 3/4	30 3/4

1983	1st	2nd	3rd	4th
High	28	33 3/16	31 7/8	33 1/8
Low	23 5/8	26 7/8	27 9/16	27 5/8

The Bank of New York Company, Inc.

48 Wall Street, New York, N.Y. 10015
(212) 530-1784

Board of Directors

H. Adams Ashforth*
Chairman of Albert B. Ashforth, Inc., real estate management and brokerage
Chairman of Ashforth Properties, Inc., real estate investment company

J. Carter Bacot *(Chrm. E)**
Chairman and Chief Executive Officer of the Company and The Bank of New York

Charles M. Bliss
Retired, formerly Chairman of The Bank of New York

John G. Brookhuis *(E)**
Vice Chairman of American Hoechst Corporation, manufacturer of chemicals

Alexander Calder, Jr.*
Chairman of the Executive Committee of Union Camp Corporation, forest products

Eli Whitney Debevoise *(E)**
Senior Partner in the firm of Debevoise & Plimpton, attorneys

Harold A. Eckmann *(E) (T)**
Chairman and Chief Executive Officer of the Atlantic Companies, property, liability and marine insurance

Ralph E. Gomory*
Senior Vice President and Director of Research, International Business Machines Corporation

Peter Herrick *(E)**
President of the Company and President and Chief Operating Officer of The Bank of New York

Lawrence Lachman *(Chrm. A) (C) (T)**
Retired, formerly Chairman and Chief Executive Officer of Bloomingdale's

Robert P. Luciano *(A) (T)**
Chairman, President and Chief Executive Officer of Schering-Plough Corporation, manufacturer of pharmaceutical and consumer products

Arthur I. Mendolia *(A)**
Chairman of CasChem, Inc., manufacturer of chemicals

Donald L. Miller *(C) (E) (S)**
Vice President, Employee Relations, Consolidated Edison Company of New York, Inc., supplier of electric, gas and steam services

H. Barclay Morley *(Chrm. S)**
Chairman and Chief Executive Officer of Stauffer Chemical Company, diversified chemical manufacturer

Edward J. Mortola *(Chrm. C) (E) (P)**
Chancellor of Pace University

John G. Phillips *(A) (P) (S)**
Retired, formerly Chairman and Chief Executive Officer of Louisiana Land and Exploration Company, producer of oil and gas

H. Ladd Plumley*
Chairman of Worcester, Mass. Redevelopment Authority

Robert T. Quittmeyer *(A) (E) (T)**
Retired, formerly Chairman and Chief Executive Officer of Amstar Corporation, manufacturer of sweeteners and industrial and technical products

Catherine A. Rein*
Vice President and General Counsel, KMI Continental, Inc., packaging, forest products, insurance, energy

Delbert C. Staley *(E) (Chrm. T)**
Chairman and Chief Executive Officer of NYNEX Corporation, telecommunications

John R. Stevenson *(E) (T)**
Chairman and Senior Partner in the firm of Sullivan & Cromwell, attorneys

Eugene J. Sullivan *(S)**
Chairman and Chief Executive Officer of Borden, Inc., producer of food, dairy and chemical products

William L. Wearly *(E) (S)**
Retired, formerly Chairman and Chief Executive Officer of Ingersoll-Rand Company, manufacturer of industrial machinery

Samuel H. Woolley *(Chrm. P)*
Retired, formerly Chairman and Chief Executive Officer of the Company

Committee Memberships
(A) Audit and Examining
(C) Community Reinvestment
(E) Executive
(P) Pension
(S) Salary and Compensation
(T) Trust and Investment
(Chrm.) Chairman of Committee

**Also a Director of The Bank of New York*

Officers

J. Carter Bacot
Chairman and Chief Executive Officer

Peter Herrick
President

Newton P.S. Merrill
Executive Vice President

Deno D. Papageorge
Executive Vice President

Joseph F. Fadell
Auditor

Robert E. Keilman
Comptroller

Robert J. Goebert
Secretary

ARCS Mortgage, Inc.
Howard J. Levine, President

Beacon Capital Management Company, Inc.
Samson Wang, President

The Bank of New York (Delaware)
John F. Barrett, President

The Bank of New York

Senior Management

J. Carter Bacot
Chairman and Chief Executive Officer

Peter Herrick
President and Chief Operating Officer

Sheldon Harrison
Senior Executive Vice President
and Chief Operations Officer

Deno D. Papageorge
Senior Executive Vice President
and Chief Financial Officer

Gregory W. Spurr, Jr.
Senior Executive Vice President
and Chief Credit and
Commercial Banking Officer

Douglas F. Adams
Executive Vice President
and Chief Trust Officer

Charles L. Booth
Executive Vice President
and Chief Investment Officer

Richard D. Field
Executive Vice President
Regional Banking

Newton P. S. Merrill
Executive Vice President
U.S. Commercial Banking

W. Alden Anderson, Jr.
Senior Vice President
Cash Management

Geoffrey W. Bennett
Senior Vice President
Personnel

Walter H. Cushman
Senior Vice President
Trust Operations

Harvey V. Delapena, Jr.
Senior Vice President
International Banking

George H. Denniston, Jr.
Senior Vice President
Special Projects

Martha L. Dinerstein
Senior Vice President
Marketing

Emmett Driscoll
Senior Vice President
Computer Services –
Securities Processing

Joseph F. Fadell
Senior Vice President and
Auditor

Robert R. Gardella
Senior Vice President
Credit

Alan R. Griffith
Senior Vice President
Communications, Entertainment
and Publishing

Joseph C. Hoopes
Senior Vice President
Personal Trust

Jerome P. Isaldi
Senior Vice President
Corporate Trust and Mutual Funds

Phillip C. Kerda
Senior Vice President
Bank Operations

Robert E. Keilman
Senior Vice President
and Comptroller

William J. Kemple
Senior Vice President
Bank Operations

Robert C. Kuner, Jr.
Senior Vice President
Securities Processing

John B. Mencke
Senior Vice President
Mortgage and Construction Lending

John E. Otis, Jr.
Senior Vice President
Northeastern Banking

Owen M. Quattlebaum
Senior Vice President
Investment

Thomas A. Renyi
Senior Vice President
Capital Markets

Leroy M. Richey, Jr.
Senior Vice President
Credit Quality Control

John A. Ross
Senior Vice President
World Asset and
Liability Management

Fordyce B. St. John, Jr.
Senior Vice President
Oil, Gas and Extractive Industry

Robert P. Ulin
Senior Vice President and
Chief Economist

Robert J. Goebert
Vice President and Secretary

Central Region

443 South Warren Street
Syracuse, N.Y. 13202
(315) 424-1000

Regional Board

David M. Beers
Crouse-Irving Memorial Proprietors, Inc

Albert C. DeFurio, M.D.
East Syracuse

Adrian P. Driggs
Fayetteville

Timothy E. Gorman
Costello, Cooney and Fearon

Keith G. Hawkins
Green Hills Farm Stores, Inc

Herbert C. Hooley
Regional President

Joseph J. Jackler
Jackler, Nellen, Liberman & Co

Gerald H. Levy
Fayetteville

David L. Osborn
Syracuse Letter Company, Inc

Stephen M. Schneeweiss
Cazenovia College

George C. Wortley, III
Manlius Publishing Corporation

Senior Management

Herbert C. Hooley
Regional President

County Region

123 Main Street
White Plains, N.Y. 10602
(914) 946-4300

Regional Board

David Bogdanoff
Montrose Construction
Company, Inc

James Dempsey
Attorney, White Plains

Ray H. dePasquale
The Technical Material Corporation

Wallace C. Doud
International Business
Machines Corporation

Edward L. Gardner
Industrial Solvents Corporation

William F. Harrington, Sr.
William F. Harrington, Attorney

R. Newton Laughlin
Rye, New York

John V. Lindholm
Guideposts Associates, Inc

Edward J. Mortola
Pace University

Paul J. Mirsky
Refined Sugars, Incorporated

Leo J. Pantas
Riverside, Conn

William G. Sharwell
Pace University

William F. Stanley
Walter A. Stanley & Son, Inc

Ross M. Weale
Regional President

Senior Management

Ross M. Weale
Regional President

Long Island Region

235 Rockaway Avenue
Valley Stream, N.Y. 11582
(516) 823-3535

Regional Board

Jeremiah J. Burns
Jeremiah J. Burns, Inc

George F. Biondo
George F. Biondo, Attorney

John J. Collins
New York Telephone

Robert Erich
Flower Time, Inc

Alexander A. Forman
Alexander A. Forman, Attorney

Robert G. Freese
Grumman Corporation

Milton A. Hendrickson
Hendrickson Bros., Inc

Paul J. Leyden
Regional President

John P. Nicholson
Nicholson & Galloway, Inc

W. Kenneth Regan
Huntington

Senior Management

Paul J. Leyden
Regional President

Mid-Hudson Region

280 Broadway
Newburgh, N.Y. 12550
(914) 561-8200

Regional Board

Monroe R. Davis
Cooke & Davis

Howard S. Finkelstein
Finkelstein, Kaplan, Levine,
Gittelsohn & Tetenbaum, PC

John J. Gartland, Jr.
Corbally, Gartland & Rappleyea

Kevin B. McGrath
Shea & Gould

Robert Parker
Concord Resort Hotel

Walter A. Rhulen
Rhulen Agency, Inc

Miles P. Shanahan
Retired

James F. Smith
Orange & Rockland Utilities, Inc

H. Kent Vanderhoef
Director of various companies

Peter Van Kleek
Regional President

Belle Mayer Zeck
Conal Ortenberg Mayer
Zeck & Prier PC

Senior Management

Peter Van Kleek
Regional President

Headquarters
48 Wall Street, New York, N.Y. 10015
(212) 530-1784

Banking Subsidiaries
The Bank of New York
48 Wall Street, New York, N.Y. 10015

The Bank of New York (Delaware)
1105 North Market Street
Wilmington, Delaware 19801

The Bank of New York Trust Company
123 Main Street
White Plains, N.Y. 10602

The Bank of New York Trust Company of Florida, N.A.
800 Brickell Avenue
Miami, Florida 33131

Non-Banking Subsidiaries
ARCS Mortgage, Inc.
7301 Topanga Canyon Boulevard
Canoga Park, California 91303

The Bank of New York Life Insurance Company, Inc.
1421 East Thomas Road
Phoenix, Arizona 85014

The Bank of New York Overseas Finance N.V.
15 Pietermaai
Curaçao, Netherlands Antilles

Beacon Capital Management Company, Inc.
48 Wall Street
New York, N.Y. 10015

BNY Financial Corporation
48 Wall Street
New York, N.Y. 10015

B.N.Y. Holdings (Delaware) Corporation
1105 North Market Street
Wilmington, Delaware 19801

BNY Leasing, Inc.
48 Wall Street
New York, N.Y. 10015

Foreign Branches of The Bank of New York
147 Leadenhall Street
London EC3V 4PN, England

Ocean Building
Collyer Quay
Singapore 1

Grand Cayman
P.O. Box 500
Cayman Islands, B.W.I.

International Banking Subsidiaries of The Bank of New York
The Bank of New York International, Incorporated
800 Brickell Avenue
Miami, Florida 33131

BNY International Investments, Inc.
800 Brickell Avenue
Miami, Florida 33131

Annual Meeting of Shareholders
The Schimmel Center for the Arts, Pace University, Spruce at William Streets, New York, N.Y., on Thursday April 11, 1985 at 10:00 a.m.

Form 10-K
A copy of the Company's Annual Report on Form 10-K to the Securities and Exchange Commission will be furnished, free of charge, to a shareholder upon written request to:

Mr. Robert J. Goebert, Secretary
The Bank of New York Company, Inc.
48 Wall Street, New York, N.Y. 10015

Listing
The Common Stock of the Company is listed on the New York Stock Exchange. Our NYSE symbol is BK. The Adjustable Rate Cumulative Preferred Stock is not currently listed.

Dividend Reinvestment Plan
Information regarding this plan will be furnished upon written request to:

Mr. Robert J. Goebert, Secretary
The Bank of New York Company, Inc.
48 Wall Street, New York, N.Y. 10015

Dividend Disbursing Agent and Transfer Agent for Stock
The Bank of New York
48 Wall Street, New York, N.Y. 10015

Registrar for Common Stock
United States Trust Company of New York
45 Wall Street, New York, N.Y. 10005

Registrar for Adjustable Rate Cumulative Preferred Stock
The Bank of New York
48 Wall Street, New York, N.Y. 10015

Independent Accountants
Deloitte Haskins & Sells
One World Trade Center
New York, N.Y. 10048

Affirmative Action Policy
The Bank of New York Company, Inc. continues its commitment to affirmative action in employment. It is the official policy of the Company to implement affirmatively equal employment opportunities for all qualified employees and applicants, regardless of race, creed, color, religion, sex, age, national origin, physical or mental handicaps or veteran status. We believe that this policy represents the most reasonable moral, social and economic point of view.

Bicentennial Book
The Bank of New York Company, Inc. recently published a history of the Bank, entitled *200 Years of Looking Ahead*, in conjunction with its Bicentennial in 1984. Shareholders and other interested persons may obtain copies by sending a check or money order for $11 per book to:
Pierre S. Brull
Director of Investor Relations
The Bank of New York Company, Inc.
48 Wall Street
New York, N.Y. 10015

Chairman FAUNTROY. Thank you, Mr. Bacot. I appreciate your testimony. I appreciate your cooperation with this subcommittee on this case. I know these are trying times for you and your employees at the bank.

While we must understand and assure ourselves the kinds of events which bring us here today are not repeated, I do want you to know that we have a great deal of empathy with the employees of the bank handling a crisis like this.

Mr. BACOT. Thank you.

Chairman FAUNTROY. You suggest that a flaw in the computer program design was at the core of this. Are you satisfied that that flaw has been corrected—that the computer will know what to do when 32,000 transactions are reached? Are you confident that that problem will not occur again?

Mr. BACOT. Yes. That problem has been corrected with a longer term redesign of the program. Whether there are other problems in there is the reason for our internal task force and the outside consultants looking into it to see if there are other problems. We don't know of any now, but we want to make sure of it.

Chairman FAUNTROY. Where does the blame lie? Have you assessed where the failure was in anticipating that problem?

Mr. BACOT. No. We have not completed the study of it yet. It is programming—someone having to do with the programming and the testing of it. Whether they were our own people, or outside consultants who worked on it, we don't know. We have not completed that investigation yet.

Chairman FAUNTROY. Do you employ software development lifecycle methodology which divides the development process into distinct phases so your managers can evaluate them?

Mr. BACOT. Mr. Papageorge.

Mr. PAPAGEORGE. What we will do is develop the software to a certain stage, test it, run it, and if it runs correctly we use that software. As we move on and develop new software, we do the same thing. There is a development period, a testing period; then we go live with the software.

Chairman FAUNTROY. Would you provide us for the record your procedures in evaluating and testing and maintaining?

Mr. PAPAGEORGE. What we would do is we would initially recognize a need, either for additional capacity or, in this case, it was an expansion of the number of securities we were holding, our recordkeeping. We would design the program, test the program, and internally, initially, to make sure they did what we wanted them to do. Then we would actually run tests using it on weekends or on nights with the Federal Reserve to make sure the interface between ourselves and the Federal Reserve works correctly. At that point, everything being tested under normal conditions with normal data, we would go live with it, presuming we had tested all the instances that could make the program fail.

Mr. BACOT. For example, we are testing an additional diagnostic system, I mentioned, right now. We will be doing it this weekend on the backup computer for testing purposes with the Federal Reserve, in conjunction with them.

Chairman FAUNTROY. Do you have a master test plan that has testing objectives you want to reach through the procedures?

Mr. **Harrison**. We do set up benchmarks for what we want the system to do with regard to volume, storage capacity. We run those tests against those benchmarks.

Chairman **Fauntroy**. I see. With respect to this particular glitch, what happened?

Mr. **Harrison**. At the time we expanded the file to 36,000 individual securities, the actual number of issues of securities trading through the system at that time was roughly 3,000. So, we had sufficient capacity and we had put in a number of—30,000-some-odd, and in another part of the program—32,000, to be exact—giving us roughly almost 10 times the capacity we needed at that point. Unfortunately, in doing that, the programming thereafter did not instruct the computers as to what to do if you breached the 32,000 level. That is what created our problem. In effect, the computer did not know what to do. In a disorganized manner, it started to accept information which corrupted the data base.

Chairman **Fauntroy**. Mr. McCollum.

Mr. **McCollum**. I think I was mispronouncing your name. I am not as good on my French as I should be.

Mr. Bacot, if you had not been able to correct this problem as rapidly as you did, although there was a delay—I know there was some discomfort to you, and pain, and some loss to your institution—but if you had not been able to correct it even within the number of hours you did, and the patch had not worked, let's say the worst case scenario, and this had gone on for—I don't know how long it could have gone on, 2 or 3 days—what would have been the consequences then to the customers, the bank, the Fed?

Mr. **Bacot**. The trading would have gone on. Settlement would have backed up. It is basically a settlement problem. It would not delay the trading, for example. The loan would stay with the Federal Reserve. We would continue to pay interest on it each day. So our costs would greatly increase. If worse came to worse and the computers would not work at all, you could put an army of people in there and eventually deliver out the securities. That would be the last gasp, if nothing else worked at all. It could go 2 to 3 days, we guess, and the system would function. Settlement would back up, but the system would function.

Mr. **McCollum**. In Mr. Corrigan's written testimony, he indicated there was some evidence that investors and, in fact, customers of yours were beginning at a point to hit the panic button on Friday morning to stop transactions that were to be cleared through your institution, or to stop dealing with those dealers who were being cleared through your institution.

Do you think that there is a lack of knowledge of the real world situation for a failure of this nature out there such that just by the size of this failure, and the word getting out on the wire services, that panic will set in in this type of setting, where a longer period of time elapses?

Mr. **Bacot**. No, I don't think so, in terms of the dealers and the participants in the business. Panic, in our opinion, did not set in. The dealers knew the situation. They cooperated fully with us and with the Federal Reserve. The Federal Reserve cooperated. All the parties, in my opinion, cooperated very well.

I think what Mr. Corrigan was referring to was the hour or so time period on Friday when they would not let deliveries come in to us. That was what caused disruption in the market. That is why they backed off it rather quickly. Had they not done that, the market would have flowed along. That was the cause of the concern not the fact that we had had a software problem.

Mr. McCOLLUM. Despite the costs in the future, you would recommend that the Fed not stop the activities like they did? Just let the costs build up until the problem is corrected?

Mr. BACOT. I think in terms of what is best for the system, to have it function effectively, we would have to do it that way without stopping.

Mr. McCOLLUM. There was a suggestion of penalty interest. You had a $5 million loss to your institution. There was some suggestion that there may be—in cases where there is sloppy work, or there is perhaps error, negligence, failure to anticipate—and maybe there was at somebody's level, whether an outside contractor for our software services or somebody in-house—there is a suggestion in those kind of situations that maybe the Fed ought to be exacting more than it would for a normal overnight loan.

What do you think of that?

Mr. BACOT. As I mentioned, I think $5 million is penalty enough in relation to our earnings, in relation to its effect on the bank's earnings, and the profit sharing of the bank in which all employees participate. There was a penalty paid for it which we think is sufficient.

Should there be an extra penalty? We say no. If there was, it would depend on how much it was. If it was very severe, we and other clearing banks would have to assess that in considering whether we wanted to continue in the business if there will be more severe penalties for computer software problems which occur from time to time.

Mr. McCOLLUM. I want to commend your institution for the innovative computer and financial services and system you have developed. I think really as we have remarked all morning—in other hearings where I have anticipated, there have been occasions to desire it—we could not have the sale of Government debt, we could not have the sale of transactions of private business in the area of Government securities, or, for that matter, most securities, if you did not have this system, this system of clearing that you have.

While I am skeptical and concerned as a citizen and as a Government official in oversight, I do want to compliment your institution on that and encourage you to continue.

Thank you, Mr. Chairman.

Chairman FAUNTROY. Mr. Bacot, this, of course, demonstrates how dependent the financial markets and indeed financial institutions are upon what these computer systems do. What management control processes you have in place to handle, and what safeguards do you have that involve your managers as against the technicians you told us about thus far?

Mr. BACOT. It is described and I think disclosed in the outline in the exhibit. We have a vice president who is a department head; another vice president over him is the group head of all of Government clearing; and a senior vice president in charge of the division.

She reports directly to Mr. Papageorge. Mr. Papageorge sits on the senior steering committee of the bank, along with Mr. Harrison, myself and two others. Thus communications go up quickly in this operation.

Chairman **Fauntroy**. Is this for all new computer operations applications that you decide to implement?

Mr. **Bacot**. We made modifications and improvements back in March.

Chairman **Fauntroy**. As a management group?

Mr. **Bacot**. That is brought up to the senior management group: Do we want to do this; spend this much on it? We spent, in the last 5 years, $6½ million to upgrade the software and computer hardware for this system alone. All that has to be approved at the senior management level before it is undertaken.

Chairman **Fauntroy**. Some people would argue the Federal Reserve discount window loan really prevented The Bank of New York from failure at this point. Are the consequences for the bank, for its managers, its stockholders, appropriately commensurate with the kind of things that were done with respect to the Continental Bank and its rescue—the type of impositions that the Federal Reserve and the FDIC proposed as part of that settlement?

Mr. **Bacot**. Mr. Chairman, I do not think there is any question of failure or insolvency of the bank at all. We didn't have cash. We had $25 billion in marketable Government securities. That is more than adequate collateral, as Mr. Corrigan and Mr. Volcker said, to cover the loan, to make it a riskless loan. There was no credit question involved here. Our customers, the dealers, were not out any money. They had the cash, not the securities. They had the ability to invest the cash, if they wanted to, overnight and make money on it. We were the only ones absorbing the cost.

It is entirely different from the Continental situation where there was a problem of credit and a problem of getting deposits. Here, we had the collateral, the securities, but we didn't have the cash. It is a matter of timing. If we had longer, we could have solved the problem in that timeframe.

Chairman **Fauntroy**. While the penalty of the $5 million was one which you paid and you were the only bank hurt by this failure, the holders of the securities had money that they were not paying to you for a while, and thus they benefited. The Fed funds rates went down during that period.

I also recall references to institutions looking to fatten their positions as a hedge against what they could not determine was either a failure or simply a technical problem. Are you confident that this kind of problem will not happen again in your facility?

Mr. **Bacot**. Let me say first, regarding the problem in the markets, it is questionable what effect or how much it had on the Federal funds rate. Normally, I think the Fed funds rate in the situation should have gone down the next day more than it did. It is a question of what effect it has on the money supply and other things. Our feeling is it was very minimal, if any, but that is something the Federal Reserve has to assess, and they are better able to comment on that than we are. I don't know the intricacies of the platinum market and what that reacts to, so I cannot comment on that.

Chairman **Fauntroy**. Thank you so much for appearing here today.

Mr. **McCollum**. I wish to thank the panel for coming, and appreciate the public interest.

Chairman **Fauntroy**. I will be sending the bank some questions at a later time as a followup.

This hearing is now adjourned.

[Whereupon, at 11:40 a.m., the subcommittee adjourned, subject to the call of the Chair.]

BLANK PAGE

204

APPENDIX

[The following material was submitted by the subcommittee for inclusion in the record:]

APPENDIX TABLE OF CONTENTS

Page

The Bank of New York response dated January 22, 1986 to Subcommittee request for additional information on November 21, 1985 computer failure 207

The Bank of New York booklet on its Government Securities Clearance Service 209

Federal Reserve Bank of New York letter of November 27, 1985 detailing developments surrounding Bank of New York operational problems of the previous week 247

House Banking letter of December 3, 1985 to the Federal Reserve requesting information on the $22.6 Discount Window Advance of November 21, 1985 to The Bank of New York 252

Response of the Federal Reserve dated January 23, 1986 to above letter 254

Loan Agreement between the Federal Reserve Bank of New York and borrowing bank 269

- -

SOFTWARE DEVELOPMENT & SECURITY DOCUMENTATION

NASA Guidelines for Assuring the Adequacy and Appropriateness of Security Safeguards in Sensitive Applications - Chapter 2: The Software Development Life Cycle & Security 273

Office of Management and Budget Circular A-71: Security of Federal Automated Information Systems . . . 340

Overview of Computer Security Certification and Accreditation (NSB Special Publication 500-109) 346

Background Information on System Development from GAO's System Audit Training Material for GAO Evaluators . . . 366

Application System Development Controls (Chapter 12) 377

Computer Software Management: A Primer for Project Management & Quality Control (NSB Special Publication 500-11) 388

Guidlines for Automatic Data Processing Risk Analysis (FIPS Publication 65) 440

Guidelines for Security of Computer Applications (FIPS Publication 73) 467

Guidelines for ADP Contingency Planning (FIPS Publication 87) 505

The Complete Guide to Software Testing Chapter 7 - Testing Systems 534

- -

Newspaper articles detailing The Bank of New York government securities computer problem 563

BEST COPY AVAILABLE

BLANK PAGE

206

THE BANK OF NEW YORK
NEW YORK'S FIRST BANK - FOUNDED 1784 BY ALEXANDER HAMILTON

48 WALL STREET, NEW YORK, N.Y. 10015

J. CARTER BACOT
CHAIRMAN OF THE BOARD

January 22, 1986

The Honorable Walter E. Fauntroy
Chairman
Subcommittee on Domestic Monetary Policy
U. S. House of Representatives
H2-109, Annex No. 2
Washington, D. C. 20515

Attention: Mr. Howard Lee
Staff Director

Dear Mr. Fauntroy:

As I discussed in my testimony before the Subcommittee on December 12, 1985, The Bank of New York was then undertaking a detailed review of its Government Securities Clearance Operation. This review was conducted by both our internal staff and special outside consultants.

In the course of this review, we have discovered additional information about the cause of the problem of November 21, 1985. Accordingly, I am writing to share with you this additional information.

As we previously described, the Bank's computer program design contained a flaw in that there were inadequate instructions regarding the storage of data when the number of securities issues to be cleared exceeded approximately 32,000. This situation occurred for the first time on November 20, 1985.

Our review disclosed that late in the clearing day of November 20, 1985, the Bank's computer issued messages which indicated that this limitation had been reached. The immediate problem was solved by deletion of certain inactive securities issues to overcome the limitation. This enabled the Bank to complete the processing of Wednesday's clearance work.

There was, however, concern by Departmental personnel that this solution was cumbersome and would hamper the Bank's ability to provide clearance services efficiently. Accordingly, the Bank's outside computer program consultants, after consultation with Departmental personnel, commenced an effort on Wednesday evening to change the computer programs to accommodate a larger number of securities issues.

The program changes were introduced during the early morning hours of Thursday, November 21 and used in the performance of a procedure referred to as "end of day processing". These changes, however, proved to be faulty and efforts to correct them during end of day processing were unsuccessful. As a consequence of the foregoing, the data base produced by end of day processing was corrupted.

I hope that this additional information further clarifies the events in question.

Best regards.

Sincerely,

J Carter Bacot

JCB:mpm

BLANK PAGE

208

BANK OF NEW YORK

GOVERNMENT SECURITIES CLEARANCE SERVICE (GSCS)

A. INTRODUCTION TO GOVERNMENT SECURITIES CLEARANCE SERVICE

The Bank of New York, with a reputation for innovative thinking and modern processing techniques regards the Government Security Clearance Service as the ultimate blend of organization and technology.

Through the Government Securities Clearance Service, The Bank of New York acts as direct or indirect clearing agent for the settlement of purchases and sales of government securities (United States Treasury Bills, Bonds, Notes and Agency Issues) among government brokers and dealers.

Unlike clearing services offered by other major government clearing agents (mainly commercial banks), The Bank of New York, through a central and solely dedicated computer system located on premises, allows participants to clear directly among themselves, thereby bypassing the Federal Reserve Bank's clearing network. A participant clearing transactions with a non-participant has the availability of a medium-speed communication link to the Federal Reserve Bank of New York. Contingent on volume, the customer also has the option of clearing directly for themselves utilizing video terminals and printers located at their own premises and linked via telephone lines to the central computer at The Bank of New York; or the Bank will provide the clearance service through video terminals located at The Bank of New York.

B. ENTERING PENDING TRANSACTIONS

The Brokers and/or Dealers enter pending transactions onto the GSCS system in any of the following ways:

- Directly onto the system via terminals located at the broker/dealer office.
- Via bulk input, utilizing magnetic tape produced by the customer's in-house computer. This tape is then delivered to the Bank and merged into the GSCS system.
- Customers who elect to have the Bank clear their transactions directly can do so by sending hard copy instructions to the Bank. Systems personnel will load these pending transactions to the GSCS system and clear the transactions directly.
- The GSCS system has the capability of establishing a computer to computer data interface for the purpose of creating receives and delivers.

These transactions are entered into the system and are stored in the system data base. Clearance transactions consist of expected receives (Receives) and deliveries to be made (Deliveries). Receives may be matched to specific Deliveries (matched trades), or multiple deliveries.

As receives come in, deliveries are made. These deliveries may be initiated by the system automatically, as in the case of matched trades, or manually by a customer operator.

C. CLEARANCE FUNCTIONS SUPPORTED BY G.S.C.S. SYSTEM

Clearance functions supported by The Bank of New York Government Securities Clearance Service may be summarized as follows:

- Clearance of securities transactions within the Bank among participants of the GSCS system.

- Clearance of securities transactions through FRBNY and acceptance of incoming transactions from FRBNY. Automated transmission and validation of messages under user control.

- Support of real time inquiry functions such as security and cash balances, message status, and pending receives and deliveries.

- Real time report generation such as fail lists, on line position reports, on line cash balances, pricing reports, and security master lists.

- End of day products, such as transaction journals, money variance reports, position reports, interest and redemption reports etc.

D. ADVANTAGES TO DIRECT USERS

The Bank's GSCS Service provides the following advantages to direct users:

- Allows Dealer/Brokers to control their own operation by clearing directly for themselves, thereby eliminating dependence on a clearing institution.

- The system provides turnaround which virtually eliminates fails and the resulting cost of carrying overnight positions.

- Allows Dealer/Brokers the opportunity to anticipate financial needs on a daily basis much more accurately due to real time inquiries during the clearing day.

- Allows Participants to clear among themselves without going through the Federal Reserve Bank of New York thereby saving the cost of Federal Reserve Bank wire charges.

- The system will stay up one half hour after the fed close to enable customers to handle internal transfers of securities between themselves.

E. ADVANTAGES TO INDIRECT USERS

The banks GSCS service provides these advantages to indirect users.

- The Bank provides individual customer contact at both the operations and administrative level, assuring personal attention and service at all times.
- The indirect user enjoys practically all the same benefits the GSCS system provides to the direct user,through direct telephone communication lines to the customer's office installed at the customer's option.
- The Bank has many years experience in clearing government securities and is currently the largest government clearing agent in the country.

F. FINANCING

All clearance accounts serviced by The Bank of New York establish lines of credit with our Wall Street Division. Primary government dealers or creditworthy secondary dealers may take positions in securities without executing deliveries. The Bank of New York will finance positions, in the form of loans to a predetermined amount, where funds are not available in the customer's deposit account. All positions are subject to standard margin requirements. If the Bank fails to turn around any item received within the system up to two minutes prior to the Fed close, the Bank will finance the fail position overnight. Items received within this two minute period and not turned around for whatever reason are subject to a customer loan.

G. PROFITABILITY

If increased profits are your goal, whether you are a Direct or Indirect user, the GSCS System will reduce your clearance costs, alleviate transaction congestion and provide the ultimate in safety, security, and privacy. The primary benefit, however, is the virtual elimination of carrying costs resulting from unnecessary fails.

H. FEES

A clearance fee is charged for the receipt and delivery of government transactions on the GSCS system. These fees are competitive within the industry as we know it today.

I. FURTHER INFORMATION

The Bank of New York has worked long and hard to produce the state of the art in computer systems for the clearing of government securities. But just as important as our technical innovations, we have maintained our strong reputation for personal service. If you would like to discuss how The Bank of New York's Government Securities Clearance Service can benefit your Company, please call or write:

Dealers and Brokers

Mr. Frederick B. Roemer
Asst.Vice President
Telephone (212) 530-8231
The Bank of New York
110 Washington Street
New York, New York 10015
Att: G.S.C.S. 3-D

Institutions

Mr Thomas Price
Vice President
Telephone (212) 530-1784
The Bank of New York
19 Rector Street
New York, New York 10015

J. SAMPLE REPORTS

The Components of the Government Security Clearance Service have been designed with you, the Customer, in mind. The following contain samples of various reports which are standard parts of the Government Security Clearance System available to our customer on a daily basis.

I. PENDING DELIVERY QUEUE

The PENDING DELIVERY QUEUE may be accessed in print or display format throughout the clearing day. The file will show the customer's current position in a particular issue and lists in descending par order all pending deliveries. The pending delivery file is constantly updated as outgoing messages are settled.

```
cusip =      TBJ2      PRINT PENDING DELIVERY QUEUE                    position
0.0000    8/26/82              5/10/          17:04                          $0

     mss num        due to party             date   dollar amount       par value   mat
     -------   ----------------------        ----   ---------------    -----------  ---
M    1300017   FIRST CHGO/DEALER             5/10      9,878,555.50     10,000,000  001
     1300018   BK OF NYC/LASSER BROS         5/10      4,705,528.13      5,000,000
H    1300019   BK OF NYC/ACLI                5/10      3,890,000.00      4,500,000
     1300020   CHASE NYC/DEALER              5/10        995,782.95      1,000,000
H    1300021   CITIBANK NYC/DEALER           5/10        998,953.12      1,000,000
M    1300022   IRVING NYC/FIRST BOSTON       5/10        978,755.50      1,000,000  002
M    1300023   MARINE NYC/CARROLL MCENT      5/10        498,795.00        500,000  003
M    1300024   BK OF NYC/FBI                 5/10        498,853.75        500,000  004
H    1300025   STATE ST BOS/CUST             5/10         95,750.00        100,000
     1300026   CHASE NYC/DEALER              5/10         97,998.00        100,000

[END OF QUEUE]
```

II. PENDING RECEIVE QUEUE

The pending receive file may be accessed in print or display format throughout the clearing day. The file shows the customer's current position in a particular issue and lists in descending par order all pending receives. The PENDING RECEIVE QUEUE is constantly updated as incoming messages are settled.

```
cusip =      TBJ2      PRINT PENDING RECEIVE QUEUE                          position
0.0000    8/26/82           5/10/         17:05                                   $0

num      due from party             date   dollar amount      par value   match num
---   -----------------------       ----  ---------------   -----------   ---------
001   IRVING NYC/DISCOUNT           5/10     9,875,000.00    10,000,000    1300017
005   IRVING NYC/RMJ                5/10     6,898,950.00     7,000,000
002   IRVING NYC/LEHMAN             5/10       975,092.50     1,000,000    1300022
003   CITIBANK NYC/DEALER           5/10       483,755.00       500,000    1300023
004   BK OF NYC/ACLI                5/10       495,000.00       500,000    1300024

[END OF QUEUE]
```

III. FAIL TO DELIVER REPORT

Delivery messages successfully entered onto the system may be listed by printing the FAIL TO DELIVER REPORT. The system constantly updates this file as deliveries are settled and only pending deliveries will appear. The report is totally on-line and is heavily utilized in analyzing positions and making strategic decisions throughout the clearing day.

```
                                FAIL TO DELIVER REPORT
                                  5/10/      17:02


SECURITY:  912794BJ2 USTR BILL  0.0000  8/26/82
POSITION:  $0.00

         MSG NUM    DUE TO PARTY                       PAR VALUE       DOLLAR AMOUNT    MATCH NUM
[MATCH] 1300017    FIRST CHGO/DEALER                 $10,000,000       $9,878,555.50       0001
        1300018    BK OF NYC/LASSER BROS              $5,000,000       $4,705,528.13
[HELD]  1300019    BK OF NYC/ACLI                     $4,500,000       $3,890,000.00
        1300020    CHASE NYC/DEALER                   $1,000,000         $995,782.95
[HELD]  1300021    CITIBANK NYC/DEALER                $1,000,000         $998,953.12
[MATCH] 1300022    IRVING NYC/FIRST BOSTON            $1,000,000         $978,755.50       0002
[MATCH] 1300023    MARINE NYC/CARROLL MCENT             $500,000         $498,795.00       0003
[MATCH] 1300024    BK OF NYC/FBI                        $500,000         $498,853.75       0004
[HELD]  1300025    STATE ST BOS/CUST                    $100,000          $95,750.00
        1300026    CHASE NYC/DEALER                     $100,000          $97,998.00

                           TOTALS ---                $23,700,000      $22,638,971.95

END OF SECURITY - 912794BJ2


SECURITY:  912810CY2 USTR BOND 14.0000 11/15/11
POSITION:  $0.00

         MSG NUM    DUE TO PARTY                       PAR VALUE       DOLLAR AMOUNT    MATCH NUM
[MATCH] 1300027    BK AMER SF/DEALER                 $10,000,000      $10,195,875.00       0001
[MATCH] 1300028    BK OF NYC/FBI                     $10,000,000      $10,905,000.00       0002
[HELD]  1300029    MARINE NYC/CARROLL MCENT           $5,000,000       $4,793,850.00
[MATCH] 1300030    IRVING NYC/DISCOUNT                $1,000,000         $999,875.00       0003
        1300031    MFRS NYC/FIRST BOSTON              $1,000,000       $1,001,500.00
[MATCH] 1300032    BK OF NYC/FBI                        $500,000         $475,950.00       0004

                           TOTALS ---                $27,500,000      $28,372,050.00

END OF SECURITY - 912810CY2

****  END OF QUEUE

                   GRAND TOTAL DELIVERY PAR VALUE ---          $51,200,000

               GRAND TOTAL DELIVERY DOLLAR AMOUNT ---      $51,011,021.95
```

IV. FAIL TO RECEIVE REPORT

Pending receive messages successfully entered onto the system may be listed by printing the FAIL TO RECEIVE REPORT. As incoming messages are settled the system constantly updates the file and only incoming fails will appear. This report is totally on-line and is utilized in the decision making process throughout the clearing day.

```
                              FAIL TO RECEIVE REPORT
                                5/10/     17:02


SECURITY:  912794BJ2 USTR BILL  0.0000  8/26/82
POSITION:  $0.00

ENTRY
NUM  DUE FROM                    DATE           PAR VALUE       DOLLAR AMOUNT  MSG NUM
0001 IRVING NYC/DISCOUNT        05/10/82      $10,000,000       $9,875,000.00  1300017
0005 IRVING NYC/RMJ             05/10/82       $7,000,000       $6,898,950.00
0002 IRVING NYC/LEHMAN          05/10/82       $1,000,000         $975,092.50  1300022
0003 CITIBANK NYC/DEALER        05/10/82         $500,000         $483,755.00  1300023
0004 BK OF NYC/ACLI             05/10/82         $500,000         $495,000.00  1300024


                        TOTALS ---            $19,000,000      $18,727,797.50

END OF SECURITY - 912794BJ2



SECURITY:  912810CY2 USTR BOND 14.0000 11/15/11
POSITION:  $0.00

ENTRY
NUM  DUE FROM                    DATE           PAR VALUE       DOLLAR AMOUNT  MSG NUM
0001 IRVING NYC/CUST            05/10/82      $10,000,000      $10,192,950.00  1300027
0002 IRVING NYC/RMJ             05/10/82      $10,000,000      $10,500,000.00  1300028
0003 CHASE NYC/DEALER           05/10/82       $1,000,000         $995,250.00  1300030
0005 FIRST CHGO/DEALER          05/10/82       $1,000,000         $950,000.00
0004 IRVING NYC/DISCOUNT        05/10/82         $500,000         $470,950.00  1300032


                        TOTALS ---            $22,500,000      $23,109,150.00

END OF SECURITY - 912810CY2

$$$$  END OF QUEUE

             GRAND TOTAL RECEIVE PAR VALUE ---          $41,500,000

          GRAND TOTAL RECEIVE DOLLAR AMOUNT ---      $41,836,947.50
```

V. ALL-FAILS

The ALL-FAILS/REPORT is an on-line listing of all transactions which are currently in a fail to receive and fail to deliver status. This report calculates total Par to be received and total funds to be paid as well as total par to be delivered and total funds to be received. It is a most useful tool in determining your cash and security position during the production day.

```
                                  ALL FAILS REPORT

SECURITY:  912794BJ2 USTR BILL  0.0000  8/26/82
POSITION:  $0.00
PENDING RECEIVE
ENTRY
NUM  DUE FROM                   DATE            PAR VALUE      DOLLAR AMOUNT  MSG NUM
0001 IRVING NYC/DISCOUNT       05/10/82       $10,000,000      $9,875,000.00  1300017
0005 IRVING NYC/RMJ            05/10/82        $7,000,000      $6,898,950.00
0002 IRVING NYC/LEHMAN         05/10/82        $1,000,000        $975,092.50  1300022
0003 CITIBANK NYC/DEALER       05/10/82          $500,000        $483,755.00  1300023
0004 BK OF NYC/ACLI            05/10/82          $500,000        $495,000.00  1300024
                            TOTALS ---        $19,000,000     $18,727,797.50

PENDING DELIVERY
        MSG NUM    DUE TO PARTY                 PAR VALUE      DOLLAR AMOUNT  MATCH NUM    DATE
(MATCH) 1300017    FIRST CHGO/DEALER          $10,000,000      $9,878,555.50     0001   05/10/82
        1300018    BK OF NYC/LASSER BROS       $5,000,000      $4,705,528.13            05/10/82
(HELD)  1300019    BK OF NYC/ACLI              $4,500,000      $3,890,000.00            05/10/82
        1300020    CHASE NYC/DEALER            $1,000,000        $995,702.95            05/10/82
(HELD)  1300021    CITIBANK NYC/DEALER         $1,000,000        $998,953.12            05/10/82
(MATCH) 1300022    IRVING NYC/FIRST BOSTON     $1,000,000        $978,755.50     0002   05/10/82
(MATCH) 1300023    MARINE NYC/CARROLL MCENT      $500,000        $498,795.00     0003   05/10/82
(MATCH) 1300024    BK OF NYC/FBI                 $500,000        $498,853.75     0004   05/10/82
(HELD)  1300025    STATE ST BOS/CUST             $100,000         $95,750.00            05/10/82
        1300026    CHASE NYC/DEALER              $100,000         $97,998.00            05/10/82
                            TOTALS ---        $23,700,000     $22,638,971.95

END OF SECURITY - 912794BJ2

SECURITY:  912810CY2 USTR BOND 14.0000 11/15/11
POSITION:  $0.00
PENDING RECEIVE
ENTRY
NUM  DUE FROM                   DATE            PAR VALUE      DOLLAR AMOUNT  MSG NUM
0001 IRVING NYC/CUST           05/10/82       $10,000,000     $10,192,950.00  1300027
0002 IRVING NYC/RMJ            05/10/82       $10,000,000     $10,500,000.00  1300028
0003 CHASE NYC/DEALER          05/10/82        $1,000,000        $995,250.00  1300030
0005 FIRST CHGO/DEALER         05/10/82        $1,000,000        $950,000.00
0004 IRVING NYC/DISCOUNT       05/10/82          $500,000        $470,950.00  1300032
                            TOTALS ---        $22,500,000     $23,109,150.00

PENDING DELIVERY
        MSG NUM    DUE TO PARTY                 PAR VALUE      DOLLAR AMOUNT  MATCH NUM    DATE
(MATCH) 1300027    BK AMER SF/DEALER          $10,000,000     $10,195,875.00     0001   05/10/82
(MATCH) 1300028    BK OF NYC/FBI              $10,000,000     $10,905,000.00     0002   05/10/82
(HELD)  1300029    MARINE NYC/CARROLL MCENT    $5,000,000      $4,793,850.00            05/10/82
(MATCH) 1300030    IRVING NYC/DISCOUNT         $1,000,000        $999,875.00     0003   05/10/82
        1300031    MFRS NYC/FIRST BOSTON       $1,000,000      $1,001,500.00            05/10/82
(MATCH) 1300032    BK OF NYC/FBI                 $500,000        $475,950.00     0004   05/10/82
                            TOTALS ---        $27,500,000     $28,372,050.00

END OF SECURITY - 912810CY2

****  END OF QUEUE

                  GRAND TOTAL DELIVERY PAR VALUE ---         $51,200,000

               GRAND TOTAL DELIVERY DOLLAR AMOUNT ---     $51,011,021.95

                   GRAND TOTAL RECEIVE PAR VALUE ---         $41,500,000

                GRAND TOTAL RECEIVE DOLLAR AMOUNT ---     $41,836,947.50
```

VI. AWAIT VALIDATION QUEUE

The AWAIT VALIDATION QUEUE lists all receives where a match does not exist or a money difference has been recognized. The customer accesses the file either by print or terminal display throughout the day to determine items he must verify. The customer has the option to reverse an unvalidated message he does not know or is unwilling to accept, or to validate the item which in effect will set the delivery in motion.

```
                    PRINT AWAIT VALIDATION QUEUE
                         4/22/         11:14

msg num    security      due from party              dollar amount       par value
-------    --------    ----------------------     ---------------     -----------
9990012   912827FH0    TOLEDO TR/TRUST                 145,197.78         150,000
9990037   912810CS5    STATE ST BOS/SPEC/TUCKER      4,876,828.13       5,000,000
9990039   912827KG6    AMER NATL CHGO/INV           10,018,576.39      10,580,000
9990042   912827KS0    CHASE NYC/CUST/THE EQUIT     26,850,251.39      26,975,000
9990044   912810CX4    CHASE NYC/CUST/STANDARD       1,230,434.67       1,000,000
9990046   912827KX9    CHASE NYC/CUST/EQUITABLE     13,165,026.39      14,000,000
9990047   912794AY0    FRB NYC/CUST/LOAN COLLAT     10,000,000.00      10,400,000
9990049   912827KW1    CHASE NYC/CUST/NY STATE       8,868,324.37       9,000,000
9990075   912827MB5    CS NATL COLA/DEALER             500,100.00         510,000
9990076   912827KW1    CS NATL COLA/DEALER             500,100.00         515,000
9990078   912827KW1    CS NATL COLA/DEALER             500,100.00         515,000
9990088   912810CY2    PROVIDENT PHIL/11667 FED     15,000,000.00      14,125,000
9990101   912810CV8    CITIBANK NYC/CUST OPPENH     20,387,784.03      19,320,000
9990102   912794BH6    CHASE NYC/CUST/THE BOARD        959,407.78       1,000,000
9990107   912810CE6    CITIBANK NYC/CUST OPPENH      9,623,674.31      14,000,000

[END OF QUEUE]
```

VII. CANCELLED MESSAGE QUEUE

A customer may print or display all messages that have been entered on the system and subsequently cancelled. The CANCELLED MESSAGE QUEUE is primarily utilized as a record of trades a customer may have "paired off" prior to activating the system for settlements.

```
                              DISPLAY CANCEL MESSAGE QUEUE
                                 5/10/        08:43

 msg num     security          due to party             dollar amount      par value
 -------     --------      ---------------------       --------------     -----------
 1300568    912827MY5     MFRS NYC/SALOMON              1,041,298.34        1,000,000
 1300573    912827MY5     MFRS NYC/SALOMON              2,082,596.69        2,000,000
 1300574    912827MY5     MFRS NYC/SALOMON              1,040,985.84        1,000,000
 1300638    912827MY5     MFRS NYC/SALOMON              2,082,127.94        2,000,000
 1300688    912827MY5     MFRS NYC/SALOMON              1,041,063.97        1,000,000
 1300690    912827MY5     MFRS NYC/SALOMON              1,040,829.59        1,000,000
 1300701    912827MY5     MFRS NYC/SALOMON              1,040,673.34        1,000,000
 1300610    912827MY5     BK OF NYC/DBLGSI                727,705.72          700,000
 1300630    912827MY5     BK OF NYC/DBLGSI              1,040,595.22        1,000,000
 1300649    912827MY5     IRVING NYC/BACHE              2,080,721.69        2,000,000
 1300624    912827MY5     IRVING NYC/LEHMAN             2,080,565.44        2,000,000
 1300634    912827MY5     IRVING NYC/LEHMAN             2,081,815.44        2,000,000
 1300622    912827MY5     IRVING NYC/DEAN WITTER        5,200,632.34        5,000,000
 1300682    912827MY5     IRVING NYC/DEAN WITTER       10,410,639.68       10,000,000
 1300730    912827MY5     IRVING NYC/DEAN WITTER        2,080,877.94        2,000,000
 1300617    912827MY5     MFRS NYC/MORGAN STANLEY       1,040,126.47        1,000,000
 1300625    912827MY5     MFRS NYC/MORGAN STANLEY       1,040,282.72        1,000,000
 1300631    912827MY5     MFRS NYC/MORGAN STANLEY       3,121,785.66        3,000,000
                                      No more entries above, but more below
```

VIII. REVERSED MESSAGE QUEUE

As a customer you may print or display the REVERSED MESSAGE QUEUE which contains all messages that have been reversed for a particular clearing day.

```
                     PRINT REVERSE MESSAGE QUEUE
                        5/10/        14:34

msg num   security   due from party              dollar amount      par value
-------   --------   ----------------------     ---------------    -----------
9990011   912827FW7  FIRST LR/TRUST/TRUST DEP          4,258.45          5,000
9990014   912827LX8  FIRST LR/TRUST/TRUST DEP        108,723.56        100,000
9990013   912827LE0  CHEMICAL NYC/CUST/CLARKE         30,022.93         30,000
9990026   912827KV3  MFRS NYC/SALOMON                      0.00      3,375,000
9990043   901178DD2  CITIBANK NYC/CUST/PRIBAN              0.00          4,000
9990074   912827MQ2  CITIBANK NYC/DEALER S123        305,702.46        300,000
9990077   912827KZ4  MFRS NYC/CUST/BEAR-10000              0.00         10,000
9990094   313311FZ0  MFRS NYC/POLLOCK                700,000.00        650,000
9990090   313311FU1  MFRS NYC/POLLOCK                104,000.00        100,000
9990089   313311FU1  MFRS NYC/POLLOCK                103,000.00        100,000
9990053   313311HC9  FCNB HOUS/CUST                1,128,513.28      1,125,000
9990046   912810CX4  IRVING NYC/SALOMON           37,251,109.85     30,000,000
9990179   313311DX7  IRVING NYC/BACHE                401,477.78        400,000
9990137   912827LB6  MFRS NYC/FIRST BOSTON         5,000,000.00      5,035,000

[END OF QUEUE]
```

IX. PRICING REPORT

A valuable tool in determining the cash value of your account for collateral loan purposes, the PRICING REPORT is also an easy reference to current security prices in the government market. The report is on-line and determines not only the market value of a customer's inventory but the margin required by The Bank of New York for financing that inventory. Current gross interest earned is also indicated, as well as the total principal and interest value directly guaranteed by the United States Government.

PRICING REPORT
5/07/   15:06

SECURITY DESCRIPTION	POSITION	05/05/05 PRICE	MARKET VALUE	MARGIN	CASH VALUE	ACCRUED INTEREST
313311MC9 FCSB AGEN 13.6500 11/01/82	1,000,000	99.84	998,440.00	19,968.80	978,471.20	2,275.00
313388HM5 FHLB AGEN 12.8000 7/25/85	100,000	95.62	95,625.00	1,912.50	93,712.50	3,626.66
313388JX9 FHLB AGEN 14.2500 4/25/89	440,000	100.00	440,000.00	8,800.00	431,200.00	2,090.00
912794BL7 USTR BILL 0.0000 9/16/82	290,000	95.23	276,193.10	1,380.96	274,812.14	0.00
912794AZ7 USTR BILL 0.0000 10/07/82	600,000	94.47	566,856.00	2,834.28	564,021.72	0.00
912794BA1 USTR BILL 0.0000 11/04/82	36,250,000	93.55	33,915,137.50	678,302.75	33,236,834.75	0.00
912794CB8 USTR BILL 0.0000 4/21/83	75,000	87.93	65,947.50	329.73	65,617.77	0.00
912810CP1 USTR BOND 10.0000 5/15/10	575,000	76.75	441,312.50	0.00	441,312.50	27,631.94
912810CV8 USTR BOND 13.8750 5/15/11	1,000,000	103.56	1,035,620.00	0.00	1,035,620.00	66,677.08
912810CY2 USTR BOND 14.0000 11/15/11	235,000	104.81	246,308.20	0.00	246,308.20	15,810.27
912827MP4 USTR NOTE 12.1250 11/30/83	20,000	97.25	19,450.00	389.00	19,061.00	1,057.56
912827NC2 USTR NOTE 13.8750 4/30/84	125,000	99.71	124,648.75	2,492.97	122,155.78	289.06
912827KY8 USTR NOTE 9.6250 8/15/85	100,000	89.06	89,062.00	1,781.24	87,280.76	2,165.62
912827NB4 USTR NOTE 16.1250 11/15/86	235,000	106.28	249,760.35	4,995.20	244,765.15	24,630.93
912827MY5 USTR NOTE 14.0000 5/15/87	100,000	100.43	100,437.00	2,008.74	98,428.26	2,061.11

TOTAL CASH VALUE	TOTAL ACCRUED INTEREST
37,939,601.73	148,315.23
GOVT. GUAR. CASH VALUE	GOVT. GUAR. ACCRUED INTEREST
37,939,601.73	148,315.23

234

X . ON LINE POSITION REPORT

The ON LINE POSITION REPORT will list in Cusip number order the current box position for your account. Opening par, received par and delivered par for that particular clearing day are also indicated. The "Par on Hold" field will indicate the par amount of securities currently pledged for whatever reason. The report is totally on-line and is constantly updated throughout the clearing day.

BEST COPY AVAILABLE

ON-LINE POSITION REPORT

5/07/ 15:00

SECURITY DESCRIPTION					OPENING PAR	RECEIVES	DELIVERS	CURRENT PAR	PAR ON HOLD
313311FK3	FCSB	AGEN	15.4500	8/02/82	10,000,000	0	10,000,000	0	0
313388JO6	FHLB	AGEN	16.5000	9/27/82	89,000,000	0	89,000,000	0	0
313586FW4	FNMA	AGEN	7.7500	3/10/83	8,615,000	0	8,615,000	0	0
901178BE2	FLBB	AGEN	8.2000	1/20/83	11,592,000	0	11,592,000	0	0
9127937H3	USTR	BILL	0.0000	5/20/82	445,000,000	19,370,000	0	464,370,000	0
912793 7J9	USTR	BILL	0.0000	6/17/82	1,487,700,000	0	15,000,000	1,472,700,000	0
912794AQ7	USTR	BILL	0.0000	5/13/82	0	7,900,000	0	7,900,000	0
912794AR5	USTR	BILL	0.0000	5/27/82	320,700,000	12,500,000	0	333,200,000	0
912794AS3	USTR	BILL	0.0000	6/03/82	77,250,000	0	0	77,250,000	0
912794AT1	USTR	BILL	0.0000	6/10/82	308,000,000	0	0	308,000,000	0
912794AU8	USTR	BILL	0.0000	6/24/82	260,200,000	0	0	260,200,000	0
912794AV6	USTR	BILL	0.0000	7/01/82	321,400,000	0	0	321,400,000	0
912794BD5	USTR	BILL	0.0000	7/08/82	521,500,000	0	0	521,500,000	0
912794AW4	USTR	BILL	0.0000	7/15/82	985,200,000	0	0	985,200,000	0
912794BE3	USTR	BILL	0.0000	7/22/82	540,000,000	59,010,000	0	599,010,000	0
912794BF0	USTR	BILL	0.0000	7/29/82	281,400,000	43,680,000	0	325,080,000	0
912794BG8	USTR	BILL	0.0000	8/05/82	996,000,000	20,025,000	163,000,000	853,025,000	0
912794AX2	USTR	BILL	0.0000	8/12/82	639,000,000	0	0	639,000,000	0
912794BH6	USTR	BILL	0.0000	8/19/82	248,000,000	0	0	248,000,000	0
912794BK9	USTR	BILL	0.0000	9/02/82	62,800,000	0	0	62,800,000	0
912794AY0	USTR	BILL	0.0000	9/09/82	537,845,000	0	313,845,000	224,000,000	0
912794BL7	USTR	BILL	0.0000	9/16/82	0	50,000,000	0	50,000,000	0
912794BM5	USTR	BILL	0.0000	9/23/82	0	50,000,000	0	50,000,000	0
912794BN3	USTR	BILL	0.0000	9/30/82	25,000,000	263,250,000	0	288,250,000	0
912794AZ7	USTR	BILL	0.0000	10/07/82	211,395,000	296,165,000	211,395,000	296,165,000	0
912827GQ9	USTR	NOTE	7.0000	5/15/82	0	9,030,000	0	9,030,000	0
912827HU9	USTR	NOTE	8.2500	6/30/82	0	51,000,000	0	51,000,000	0
912827JV5	USTR	NOTE	9.0000	8/15/82	0	60,000,000	0	60,000,000	0
912827HB1	USTR	NOTE	7.1250	11/15/82	0	153,000,000	0	153,000,000	0
912827FH0	USTR	NOTE	8.0000	2/15/83	0	22,296,000	0	22,296,000	0
912827HG8	USTR	NOTE	7.8750	5/15/83	0	49,000,000	0	49,000,000	0
912827JT0	USTR	NOTE	8.8750	6/30/83	0	12,500,000	0	12,500,000	0
912827KR2	USTR	NOTE	9.2500	8/15/83	58,000,000	0	58,000,000	0	0
912827MF6	USTR	NOTE	16.2500	8/31/83	12,170,000	12,170,000	12,170,000	12,170,000	0
912827MP4	USTR	NOTE	12.1250	11/30/83	13,000,000	13,000,000	13,000,000	13,000,000	0
912827MR0	USTR	NOTE	13.0000	12/31/83	195,500,000	195,500,000	195,500,000	195,500,000	0
912827MZ2	USTR	NOTE	14.1250	3/31/84	32,550,000	70,000,000	32,550,000	70,000,000	0
912827MC2	USTR	NOTE	13.8750	4/30/84	74,300,000	29,580,000	74,300,000	29,580,000	0

TOTAL SECURITY BALANCE	9,064,126,000
TOTAL SECURITIES ON HOLD	0
AVAILABLE PAR	9,064,126,000

236

XI. TRANSACTION JOURNAL

The TRANSACTION JOURNAL is an end-of-day product which will list in Cusip number order all pertinent information on every settled activity for the previous day. It is considered the ultimate aid in locating any proof differences you may encounter and can be stored as a permanent record for auditing purposes.

```
                         SECURITIES CLEARANCE SYSTEM

                             TRANSACTION JOURNAL
                                  19-FEB-8
                                FINAL CLOSING

SECURITY - 912794BH6 USTR BILL  0.0000  8/19/82

OPENING BALANCE -              $0.00

TYPE      MSG NUM               PAR VALUE           DOLLAR AMOUNT    CONTRA PARTY
----      -------               ---------           -------------    ------------
DELV      0500001          $10,000,000.00           $9,292,591.67    CHEMICAL NYC/DEALER
REC       9990002          $10,000,000.00           $9,278,800.17    BK OF NYC/ACLI

CLOSING BALANCE -              $0.00


SECURITY - 912794AZ7 USTR BILL  0.0000 10/07/82

OPENING BALANCE -              $0.00

TYPE      MSG NUM               PAR VALUE           DOLLAR AMOUNT    CONTRA PARTY
----      -------               ---------           -------------    ------------
DELV      0500004          $10,000,000.00           $9,200,000.00    CHEMICAL NYC/DEALER
REC       9990003          $10,000,000.00           $9,150,000.00    BK OF NYC/HUTTON

CLOSING BALANCE -              $0.00


SECURITY - 912827MV1 USTR NOTE 14.6250  2/15/85

OPENING BALANCE -              $0.00

TYPE      MSG NUM               PAR VALUE           DOLLAR AMOUNT    CONTRA PARTY
----      -------               ---------           -------------    ------------
DELV      0500002          $10,000,000.00           $9,968,370.17    BK OF NYC/GOLDMAN
DELV      0500003           $5,000,000.00           $4,996,685.08    MFRS NYC/BRIGGS
REC       9990001           $5,000,000.00           $4,970,803.06    BK OF NYC/ACLI
REC       9990004          $10,000,000.00           $9,999,620.17    BK OF NYC/CUST

CLOSING BALANCE -              $0.00
```

238

XII. FINAL CLOSING POSITION REPORT

The FINAL CLOSING POSITION REPORT is an end-of-day product and is identical in format to the on-line position report, indicating opening par, receives, delivers, closing par, and par on hold in Cusip number sequence.

BEST COPY AVAILABLE

SECURITIES CLEARANCE SYSTEM

POSITION REPORT
06-MAY-
FINAL CLOSIN

SECURITY DESCRIPTION					OPENING PAR	RECEIVES	DELIVERS	CLOSING PAR	PAR ON HOLD
313311FK3	FCSB	AGEN	15.4500	8/02/82	0	10,000,000	0	10,000,000	0
313388JG6	FHLB	AGEN	16.5000	9/27/82	0	89,000,000	0	89,000,000	0
313586FW4	FNFM	AGEN	7.7500	3/10/83	0	8,615,000	0	8,615,000	0
901178BE2	FLBB	AGEN	8.2000	1/20/83	0	11,592,000	0	11,592,000	0
9127937H3	USTR	BILL	0.0000	5/20/82	460,000,000	0	15,000,000	445,000,000	0
9127937J9	USTR	BILL	0.0000	6/17/82	1,494,200,000	50,000,000	56,500,000	1,487,700,000	0
912794AR5	USTR	BILL	0.0000	5/27/82	390,700,000	0	70,000,000	320,700,000	0
912794AS3	USTR	BILL	0.0000	6/03/82	77,250,000	0	0	77,250,000	0
912794AT1	USTR	BILL	0.0000	6/10/82	358,000,000	0	50,000,000	308,000,000	0
912794AU8	USTR	BILL	0.0000	6/24/82	306,880,000	0	46,680,000	260,200,000	0
912794AV6	USTR	BILL	0.0000	7/01/82	321,400,000	0	0	321,400,000	0
912794BD5	USTR	BILL	0.0000	7/08/82	521,500,000	0	0	521,500,000	0
912794AW4	USTR	BILL	0.0000	7/15/82	985,200,000	0	0	985,200,000	0
912794BE3	USTR	BILL	0.0000	7/22/82	570,000,000	0	30,000,000	540,000,000	0
912794BF0	USTR	BILL	0.0000	7/29/82	346,400,000	0	65,000,000	281,400,000	0
912794BG8	USTR	BILL	0.0000	8/05/82	233,000,000	855,000,000	92,000,000	996,000,000	0
912794AX2	USTR	BILL	0.0000	8/12/82	589,000,000	65,000,000	15,000,000	639,000,000	0
912794BH6	USTR	BILL	0.0000	8/19/82	233,000,000	15,000,000	0	248,000,000	0
912794BK9	USTR	BILL	0.0000	9/02/82	323,990,000	0	261,190,000	62,800,000	0
912794AY0	USTR	BILL	0.0000	9/09/82	224,000,000	313,845,000	0	537,845,000	0
912794BN3	USTR	BILL	0.0000	9/30/82	288,830,000	0	263,830,000	25,000,000	0
912794AZ7	USTR	BILL	0.0000	10/07/82	248,720,000	211,395,000	248,720,000	211,395,000	0
912794CB8	USTR	BILL	0.0000	4/21/83	170,480,000	0	170,480,000	0	0
912827JT0	USTR	NOTE	8.8750	6/30/83	211,080,000	0	211,080,000	0	0
912827KR2	USTR	NOTE	9.2500	8/15/83	0	58,000,000	0	58,000,000	0
912827MF6	USTR	NOTE	16.2500	8/31/83	0	12,170,000	0	12,170,000	0
912827MP4	USTR	NOTE	12.1250	11/30/83	0	13,000,000	0	13,000,000	0
912827MR0	USTR	NOTE	13.0000	12/31/83	0	195,500,000	0	195,500,000	0
912827MZ2	USTR	NOTE	14.1250	3/31/84	0	32,550,000	0	32,550,000	0
912827NC2	USTR	NOTE	13.8750	4/30/84	0	74,300,000	0	74,300,000	0
TOTALS					8,353,630,000	2,014,967,000	1,595,480,000	8,773,117,000	0

XIII. CASH SUMMARY

The CASH SUMMARY report simply shows gross cash received, gross cash paid and the net to close for items transacted on the government wire.

SECURITIES CLEARANCE SYSTEM

29-MAR-
FINAL CLOSING

RECEIVES	DELIVERS	BALANCE
$1,777,778,136.17	$1,772,562,966.95	$5,215,169.22

XIV. A DUE TO MASTER FILE

A DUE TO MASTER FILE will be created on an individual customer basis and will be uniquely tailored to your customer base.

DUE TO FILE REPORT
2/02/8 13:06

DUE TO CODE	ABA NUMBER	DUE TO NAME	
HW18	062200424	ALA MONTG/TRUST	
ALAMO	114000022	ALAMO SA	
AAAAA	125200206	ALASKA NATL FAIR/INV	
ALBQ	107000327	ALBQ NATL	
8L080	107000327	ALBQ NATL/INV	
FU37	101200039	AM NATL STJ/	
1L227	107000932	AMER ALBQ/CUST	
AMAUS	114900180	AMER AUSTIN	
AMACU	114900180	AMER AUSTIN/CUST	
AMAIN	114900180	AMER AUSTIN/INV	
AMATR	114900180	AMER AUSTIN/TRUST	
ABLTR	072400670	AMER BK LANSING/TRUST	
8Y266	072400670	AMER BK LANSING/TRUST/S/K NBR 0067	
8F099	061300419	AMER CHAT/CUST/CRYSTAL CO REVISED PENS PL.	
AMCDL	061300419	AMER CHAT/DEALER	
8D034	061300419	AMER CHAT/INV	
MURBT	061300419	AMER CHAT/MURFREESBORO	
MJ110	061300419	AMER CHAT/TRUST/3990	
1S012	061300419	AMER CHAT/TRUST/TRASTATE CARPENTERS PENS TR FD/D	
AMFL	074000010	AMER FLET INDPLS/ANDREW LANZAROTTA RETAIL CLERKS LOCAL 1099	
LM119	074000010	AMER FLET INDPLS/CUST/BUTLER UNIV HOLCOMB INST PERM REV AC	31350

244

XV. INTERNAL AND REDEMPTION REPORT

For securities held in your position as of close of business today, the INTEREST AND REDEMPTION REPORT is an end-of-day product which determines the coupon interest and redemptions due your account the next business day. It simplifies the reconcilement of coupon interest and redemption for both the customer and the Bank by listing in a brief format each issue with the interest and redemption amount due.

BEST COPY AVAILABLE

```
                                   SECURITIES CLEARANCE SYSTEM

                                 INTEREST AND REDEMPTIONS REPORT
                                            24-FEB-8
                                          FINAL CLOSING
            INTEREST SECTION


--------------SECURITY--------------           POSITION       INTEREST DUE
313388EU0 FHLB AGEN  8.6250  2/25/82        $280,000.00         $12,075.00
313388FP0 FHLB AGEN  7.9500  2/25/82      $3,010,000.00        $119,647.50
313388HL7 FHLB AGEN 13.1000  8/25/82        $120,000.00          $7,860.00
313388GS3 FHLB AGEN 11.2500  8/25/82        $170,000.00          $9,562.50
313388JK7 FHLB AGEN 15.5500  2/25/83         $50,000.00          $2,570.05
313388GF1 FHLB AGEN  9.0000  2/25/83        $140,000.00          $6,300.00
313388GY0 FHLB AGEN 12.2500  8/25/83        $800,000.00         $49,000.00
313388GJ3 FHLB AGEN  9.3000  8/25/83      $1,785,000.00         $83,002.50
313388FE5 FHLB AGEN  7.3750  2/25/85         $40,000.00          $1,475.00
313388GN4 FHLB AGEN  9.5500  2/25/86         $20,000.00            $955.00

    ** TOTAL INTEREST DUE                                      $292,447.55


    REDEMPTIONS SECTION


--------------SECURITY--------------           POSITION      PRINCIPAL DUE
313388EU0 FHLB AGEN  8.6250  2/25/82        $280,000.00        $280,000.00
313388FP0 FHLB AGEN  7.9500  2/25/82      $3,010,000.00      $3,010,000.00
912793BG4 USTR BILL  0.0000  2/25/82      $2,520,000.00      $2,520,000.00

    ** TOTAL REDEMPTIONS DUE                                 $5,810,000.00

    ** TOTAL MONEY DUE                                       $6,102,447.55
```

246

BOARD OF GOVERNORS
OF THE
FEDERAL RESERVE SYSTEM
WASHINGTON

November 27, 1985

PRESTON MARTIN
VICE CHAIRMAN

The Honorable Walter E. Fauntroy
Chairman
Subcommittee on Domestic Monetary Policy
Committee on Banking, Finance and
  Urban Affairs
House of Representatives
Washington, D.C. 20515

Dear Chairman Fauntroy:

I am sure by this time you are aware that the Bank of New York experienced serious operational problems late last week which required the Federal Reserve Bank of New York to extend to them a sizable discount window advance.

Enclosed is a letter from E. Gerald Corrigan, President of the Federal Reserve Bank of New York, which explains in some detail the circumstances surrounding the problem and the advance. This occurrence underscores the need for the Federal Reserve to take all necessary and reasonable steps to ensure the safety, soundness, and integrity of the payments system.

In the Chairman's absence, I shall keep you informed of any additional developments and would be happy to answer any additional questions you may have.

Sincerely,

Preston Martin

Enclosure

BLANK PAGE

FEDERAL RESERVE BANK OF NEW YORK
NEW YORK, N.Y. 10045
AREA CODE 212 791-5173

E. GERALD CORRIGAN
PRESIDENT

November 27, 1985

The Honorable Preston Martin
Vice Chairman
Board of Governors of the
Federal Reserve System
Washington, D.C. 20551

Dear Pres:

Following up on our discussion of the past several days, I want to summarize the developments surrounding Bank of New York's ("BONY") operational problem of last week and our response to it.

We became aware of a potentially serious computer systems problem at BONY in the early afternoon of Thursday, November 21. As you know, reports of operational problems from major participants--including the Federal Reserve Banks--in the securities clearance network are not uncommon. Normally, these problems are corrected within a matter of hours or less, and BONY gave us assurances that they would be back in operation within a reasonable period of time. These reassurances were repeated frequently over the course of the day and into the night.

In the meantime, BONY was receiving large quantities of securities in book-entry form the payment for which was being made by automatic debits to its reserve account. In the normal course, these securities would have been redelivered to others by BONY for payment had its systems been working. However, since BONY was unable to redeliver these securities and receive payment from others, these debits continued to build during the course of the normal business day for effecting such transactions. In this regard the "normal" business day for securities transfers via the book-entry system ends at 2:30 p.m., although that cut-off hour is extended for limited times with great regularity. In fact, traffic in the market is now so large and processing backlogs so frequent that there have been only two business days this year in which we have been able to close the securities wire at 2:30 p.m. It is also worth noting that large banks, and especially large clearing banks, typically build up overdraft positions with us during the day. Indeed, it is precisely this phenomenon that has prompted the Federal Reserve to devote so much energy and attention to the subject of daylight overdrafts over the past several years.

By late afternoon, it appeared to us--BONY's reassurances notwithstanding--that their problem might not be resolved within the next few hours. Faced with that possibility--which still seemed the unlikely outcome--we began, as a matter of contingency, preparing special loan agreements which would permit the Federal Reserve Bank of New York to obtain collateral coverage for a very sizable advance via the discount window in the event such an extraordinary overnight lending facility was needed. We did so keeping in mind that the structure of the payment and reserve accounting system at the Federal Reserve Bank of New York, and the structure of payment and customer accounting systems at commercial banks, does not permit indefinite extensions. It is necessary to close down the systems at some point in order to allow time for end-of-day accounting, reconciliation and preparation and delivery of account statements for banks and other market participants all over the country before the opening of business the next day. Indeed, a failure to get the end-of-day processing done entails the risk of more generalized problems the next day.

Throughout the afternoon and into the evening we were monitoring the level of accumulated debits in BONY's reserve account. By early evening, prospects for recovery were still unclear but the passage of time had rendered the situation that much more serious. Faced with this and the very sizable overdraft in BONY's account, we requested BONY to execute the special loan agreements referred to above, even though we were still hopeful that the problem would be solved. Under those agreements all of BONY's domestic assets and certain securities that they were able to repledge were provided as collateral for any discount window credit to BONY that night.

By mid evening, BONY was able to begin sending a limited volume of security transactions using a patched up system. Bearing in mind the need to prepare for the next day's business, we kept the wires opened for as long as possible, finally closing the security wire at 1:30 a.m. Friday, when it was abundantly clear that BONY could not restore normal operations and other participants were encountering problems keeping their systems open at such a late hour. To permit BONY to buy funds to further reduce its borrowing need we extended the funds wire until 2:15 a.m. Friday. Finally, when all wires were closed a discount window advance of $22.6 billion was extended to BONY.

The advance to BONY was made at the basic discount rate of 7.5 percent. As it materialized, final accounting and reconcilement of the day's work also revealed that BONY incurred a $1 billion overdraft in its reserve account. The Federal Reserve Bank of New York will be compensated for the overdraft at the customary penalty rate of the effective Federal funds rate plus 2 percentage points. In addition, and in accordance with Federal Reserve policy, BONY will, in effect, incur some expense in maintaining extra reserve balances over the current reserve maintenance period to "make up" the $1 billion reserve account overdraft on November 22.

Of the many issues raised by this episode, the most obvious relates, of course, to the size of the overdraft built up by BONY on Thursday. Given the time of the day we learned of a potentially serious problem at BONY and given repeated reassurances and other experience with similar situations, it seemed only reasonable to proceed on the assumption that they would indeed be operational well before the constraint of end-of-day processing requirements would force us to close the wires notwithstanding BONY's position. Indeed, while processing problems in the overall system are common, it was the first instance in which a major participant was not able to close out its affairs before midnight.

Nevertheless, the question naturally arises as to whether we all might have been better off had incoming deliveries of securities to BONY been shut off at the first sign of a problem on Thursday. Such a cutoff runs the risk of generalized disruptions in the market as suggested by the experience of the next day when, around midday, we temporarily stopped accepting securities transfers for the account of BONY. The purpose of this temporary stoppage was to attempt to stabilize the situation somewhat and to see whether it was practical to prevent further increases in the overdraft without causing excessive market disruption. This temporary stoppage resulted--in quick order--in a back up in the willingness and ability of other market participants to transfer securities among themselves raising the specter of a much larger and more disruptive situation. Shortly after the stoppage was imposed, however, BONY informed us that it had regained normal operational capability. Accordingly, the stoppage was lifted at about 1:30 p.m. in circumstances in which it was clear that BONY was operational.

By early evening the backlog at BONY and in the market generally was eliminated and all of the wires closed at 8:45 p.m. By this time BONY has repaid its advance, and it had a net positive balance in its reserve account as of the close of business Friday.

Needless to say, this episode dramatically underscores the importance of the efforts already underway in the Federal Reserve and in financial institutions generally to take all reasonable steps to insure that the safety and integrity of the payments system is preserved and enhanced. We will keep you informed of further developments.

Sincerely,

Jerry

E. Gerald Corrigan
President

FERNAND J. ST GERMAIN, RHODE ISLAND, CHAIRMAN
HENRY B. GONZALEZ, TEXAS
FRANK ANNUNZIO, ILLINOIS
PARREN J. MITCHELL, MARYLAND
WALTER E. FAUNTROY, DISTRICT OF COLUMBIA
STEPHEN L. NEAL, NORTH CAROLINA
CARROLL HUBBARD, JR., KENTUCKY
JOHN J. LAFALCE, NEW YORK
STAN LUNDINE, NEW YORK
MARY ROSE OAKAR, OHIO
BRUCE F. VENTO, MINNESOTA
DOUG BARNARD, JR., GEORGIA
ROBERT GARCIA, NEW YORK
CHARLES E. SCHUMER, NEW YORK
BARNEY FRANK, MASSACHUSETTS
BUDDY ROEMER, LOUISIANA
RICHARD H. LEHMAN, CALIFORNIA
BRUCE A. MORRISON, CONNECTICUT
JIM COOPER, TENNESSEE
MARCY KAPTUR, OHIO
BEN ERDREICH, ALABAMA
SANDER M. LEVIN, MICHIGAN
THOMAS R. CARPER, DELAWARE
ESTEBAN E. TORRES, CALIFORNIA
GERALD D. KLECZKA, WISCONSIN
BILL NELSON, FLORIDA
PAUL E. KANJORSKI, PENNSYLVANIA
BART GORDON, TENNESSEE
THOMAS J. MANTON, NEW YORK
JAIME B. FUSTER, PUERTO RICO

U.S. HOUSE OF REPRESENTATIVES
COMMITTEE ON BANKING, FINANCE AND URBAN AFFAIRS
NINETY-NINTH CONGRESS
2129 RAYBURN HOUSE OFFICE BUILDING
WASHINGTON, DC 20515

CHALMERS P. WYLIE, OHIO
STEWART B. McKINNEY, CONNECTICUT
JIM LEACH, IOWA
NORMAN D. SHUMWAY, CALIFORNIA
STAN PARRIS, VIRGINIA
BILL McCOLLUM, FLORIDA
GEORGE C. WORTLEY, NEW YORK
MARGE ROUKEMA, NEW JERSEY
DOUG BEREUTER, NEBRASKA
DAVID DREIER, CALIFORNIA
JOHN HILER, INDIANA
THOMAS J. RIDGE, PENNSYLVANIA
STEVE BARTLETT, TEXAS
TOBY ROTH, WISCONSIN
ROD CHANDLER, WASHINGTON
AL McCANDLESS, CALIFORNIA
JOHN E. GROTBERG, ILLINOIS
JIM KOLBE, ARIZONA
J. ALEX McMILLAN, NORTH CAROLINA

(202) 225-4247

December 3, 1985

Honorable Paul A. Volcker
Chairman, Board of Governors
Federal Reserve System
20th and Constitution Avenue, N.W.
Washington, D.C. 20551

Dear Mr. Chairman:

During the Financial Institutions Subcommittee markup of the Expedited Funds Availability Act on November 13, I attempted to obtain adoption of an amendment enhancing federal supervision and regulation of the payments system. I was not successful in my efforts and ultimately had to withdraw my amendment. As I did so, I said to my colleagues, "I hope that in placating those who have raised such a storm of protest over this amendment, we don't pick up a newspaper in the near future and find that the potential abuses and problems that have been cited to us, have occurred."

One week later I read in the papers that the Federal Reserve had to make a $22.6 billion Discount Window Advance to the Bank of New York for a major payments system "snafu" of the sort that motivated the concerns I expressed in the Subcommittee markup.

To help me and my colleagues fully appreciate the nature and magnitude of the risks evidenced by the Manufacturers Hanover and Bank of New York situations and by the amounts of funds the Federal Reserve had to advance in order to resolve them, please provide reponses to the following questions:

1. How does the amount advanced by the Discount Window to the Bank of New York compare with the amount the Federal Reserve had to lend to the Continental Illinois National Bank in mid-1984?

BEST COPY AVAILABLE

2. If the Federal Reserve had been unable to resolve the Bank of New York situation as quickly as it did or had there been on the same day several such situations, what might the effects have been on:

- a. the domestic payments system,
- b. the domestic banking system generally,
- c. the Federal Reserve System, and
- d. the international payments and banking systems.

More generally, does the national security of our country depend in part on the stability of our payments systems?

3. With regard to the rate charged on the Discount Window Advance to the Bank of New York:

a. Was the rate charged below the prevailing market rate on overnight credit such as the Federal Funds rate? If so, how is this subsidy justified, and what incentives does it provide to banks to correct their operating patterns?

b. Is the interest rate expense the Bank of New York incurred on the Discount Window Advance tax deductible? Assuming that it is, does this not mean that, ultimately, the cost of the advance will be borne by American taxpayers?

4. Does the Federal Reserve have a set of operating, security, and financial soundness standards that depository institutions must meet before they can enter into large dollar and large security transfer transactions? If the Federal Reserve has such standards, do they include examination, monitoring, and testing of computer system security and backup systems? More specifically with regard to the Bank of New York situation, what standards must "dealer banks" meet in order to engage in large dollar security transfer activities?

5. Please provide a listing of the dates and amounts of "daylight overdrafts" over $1 million incurred by U.S. banks since January 1, 1985.

6. In view of the comments of Federal Reserve Bank President Gerald Corrigan in his correspondence to Vice Chairman Preston Martin and your likely responses to question 2 above, I am concerned about the apparent concentration of responsibility for government securities transactions among the thirty-five or so "primary dealers". Are we not courting Continental-scale problems by permitting financial operations of such enormous magnitude and significance to be handled by so few institutions? Would there not be some risk-reducing, diversification benefits from limiting the responsibilities of any single institution and increasing the number of institutions involved?

Your attention to the questions raised above and any explanatory information you feel we should have, will be very much appreciated.

Sincerely,

Fernand J. St Germain
Chairman

BEST COPY AVAILABLE

BOARD OF GOVERNORS
OF THE
FEDERAL RESERVE SYSTEM
WASHINGTON, D. C. 20551

PAUL A. VOLCKER
CHAIRMAN

January 23, 1986

The Honorable Fernand J. St Germain
Chairman
Committee on Banking, Finance and
Urban Affairs
House of Representatives
Washington, D.C. 20515

Dear Chairman St Germain:

Thank you for your letter of December 3 asking a number of questions related to the discount window loan of November 21 to the Bank of New York (BONY) and the situation that gave rise to that loan. In addition to the answers to your questions contained in the enclosed staff memorandum, I have enclosed the statements by President Corrigan of the Federal Reserve Bank of New York (FRBNY) and myself before the Subcommittee on Domestic Monetary Policy on December 12; these statements contain a detailed account of the incident and discuss many of the major public policy issues it has raised.

I hope this information is useful to your Committee.

Sincerely,

Paul

Enclosures

January 1986

Staff Memorandum Responding to Questions from Chairman St Germain Related to the Discount Window Loan to Bank of New York on November 21, 1985

Question 1. How does the amount advanced by the Discount Window to the Bank of New York compare with the amount the Federal Reserve had to lend to the Continental Illinois National Bank in mid-1984?

Answer. The circumstances facing BONY and Continental Illinois National Bank were quite different, and this is reflected in their borrowings at the discount window. The loan to BONY was a result of a computer software failure at the bank which prevented BONY from redelivering about $25 billion in U.S. government securities which it had received as clearing agent for various brokers, dealers, and other participants in the government securities market. The BONY borrowing was for one day only, and enabled this institution to meet a sudden, one-time need for liquidity. That need was eliminated once the institution was able to resend securities in question. In contrast, Continental borrowed over an extended period because it faced a loss of confidence in financial markets which rendered it unable to raise funds needed to sustain its normal operations, and which could only be restored through a major restructuring of the bank. With respect to amount, the BONY loan of $22.6 billion was well in excess of any single-day loan to Continental, whose borrowings peaked at over $7 billion in the summer of 1984, but over time Continental borrowed far more than BONY.

Question 2. If the Federal Reserve had been unable to resolve the Bank of New York situation as quickly as it did or had there been on the same day several such situations, what might the effects have been on:

a. the domestic payments system,
b. the domestic banking system generally,
c. the Federal Reserve System, and
d. the international payments and banking systems.

More generally, does the national security of our country depend in part on the stability of our payments systems?

Answer. In this instance, given the manner in which the BONY situation was handled, the problems of that individual bank did not appreciably affect the domestic and international payments system more generally. However, as noted in the material submitted for the December 12 hearings, the incident underscores the need to deal with longer-term concerns about the reliability and integrity of large dollar funds and securities transfer systems--concerns which the Federal Reserve has been actively encouraging participants to address.

Initially, while BONY was accepting securities against payment, but not sending securities and receiving funds--and was thus building a large overdraft position at the Federal Reserve--there was virtually no fallout on the financial system. For the most part, while the problem was being worked on, other market participants perceived it as a temporary, technical difficulty that did not pose any threat to the viability of BONY or its customers. The possibility of broader disruption began to emerge for a short period on Friday when, in the face of continued uncertainty about the ability of BONY to make its system operational in a timely manner and to control its overdraft position, BONY no longer was allowed to accept and make payments for securities it was receiving. This period was quite limited--BONY's full access to the securities system was restored as soon as it could process transactions--but during that time some other clearing banks and dealers curtailed their own operations in government securities markets. This reaction demonstrates the virtual impossibility of "disconnecting" a major participant in this highly automated and interrelated market, and suggests the potential for larger and more disruptive problems should market participants for any reason begin to doubt their ability to clear government securities over this system in a safe, assured fashion.

Confidence in the payments system is central to the confidence in and the efficiency of our economic system as a whole. While it is difficult to speculate on the specific effects had BONY's problem persisted or had other key institutions been involved, there would clearly have been the potential for some markets to shut down and for uncertainties and business disruption to spread more broadly into other markets. However, one of the main functions of a central bank, clearly, is to help assure that such disruptions do not occur, which is one aspect of its role as lender of last resort. But it is equally important to minimize the potential for involving central bank funds, with the reverberations and doubts about the market's ability to function on its own that may be entailed. In that regard, the Federal Reserve is stepping up its own efforts, and its work with the industry, to achieve more reliable backup systems and to reduce day-to-day levels of risks in the payments system.

Question 3. With regard to the rate charged on the Discount Window advance to the Bank of New York:

a. Was the rate charged below the prevailing market rate on overnight credit such as the Federal Funds rate? If so, how is this subsidy justified, and what incentives does it provide to banks to correct their operating patterns?

b. Is the interest rate expense the Bank of New York incurred on the Discount Window advance tax deductible? Assuming that it is, does this not mean that, ultimately, the cost of the advance will be borne by American taxpayers?

Answer. BONY borrowed at the basic discount rate of 7-1/2 percent, about 1/2 of a percentage point below the federal funds rate that day. However, as it happened, other payments made by BONY directly or indirectly to the FRBNY--owing to penalties assessed for an overnight overdraft in its reserve account and the costs of failure to deliver securities to the FRBNY in conjunction with certain open market transactions--more than made up the difference between the discount and market rates.

In any event, in the situation it faced, a lower cost of funds would not have provided BONY with a profit opportunity, but would only have served to reduce slightly its out-of-pocket costs. The discount window borrowing was needed to fund temporary holdings of securities by BONY, but the interest earned on the securities was passed on to those dealers and investors that had been anticipating delivery. Thus, any borrowing costs, even if lower than the federal funds rate, represented a net cost to BONY. One would expect that the potential for such costs would provide a significant incentive to clearing banks to have computer systems that are less prone to such failures. Even so, the Federal Reserve recognizes that, given the potential consequences of a payments system problem, additional encouragement to build safeguards into operational systems might be in the public interest. Consequently, we are looking into various options for determining the interest charge, including placing it at a penalty, when significant amounts of funds are advanced to institutions that would otherwise be in an overdraft position at the end of the day because of the failure of their computer system. We are also considering what more might be done through the bank supervisory process to provide countervailing measures to the intense competitive pressures to minimize costs and cash balances.

As with any business expense, the BONY interest payment to the FRBNY is deductible from its income before computing taxes, and with BONY realizing no compensating gains from this incident, both its income and income tax payment will be lower. However, it is not possible to conclude that there will be a net cost to the taxpayers as a result, since some other parties were presented with opportunities for gains in income and will have associated higher tax payments. These would include investors who might have earned interest on funds that otherwise would have been transferred to BONY in exchange for securities, or banks in which such funds may have been held as demand deposits.

Question 4. Does the Federal Reserve have a set of operating, security, and financial soundness standards that depository institutions must meet before they can enter into large dollar and large security transfer transactions? If the Federal Reserve has such standards, do they include examination, monitoring, and testing of computer system security and backup systems? More specifically with regard to the Bank of New York situation, what standards must "dealer banks" meet in order to engage in large dollar security transfer activities?

Answer. Under the terms of the Monetary Control Act of 1980, access to the large dollar funds and securities transfer services offered by the Federal Reserve is available to all depository institutions. Access to similar systems sponsored by private sector entities is determined by criteria set by the sponsors, but with a general expectation that those criteria be reasonable and applied in a non-discriminatory fashion.

While the Federal Reserve does not have a detailed set of operational, security, and financial soundness standards that depository institutions must meet to engage in these services, for some years the Federal Reserve has been actively encouraging payments system participants to adopt measures and policies to limit risks in such systems. Following long discussions with interested parties, the Board issued in May of 1985 a policy statement calling upon participants in large dollar payments systems to introduce measures to control and reduce risks across all such systems, including Fedwire. It is our general practice to place especially tight controls on the amount of so-called Fedwire "daylight overdrafts" which may be incurred by those institutions whose overall financial condition is considered to pose special risk.

At the bank supervisory level, since 1980 all federal bank supervisory agencies follow uniform procedures for examinations of automated data centers supporting the operations of depository institutions. These so-called "EDP examinations" focus primarily on the effectiveness of provisions for internal control and include reviews of audit coverage, management supervision, systems development and maintenance standards, as well as computer operations.

More specifically with regard to supervision of securities clearing operations, since 1982 the Federal Reserve has had special guidelines for examiners' use which are keyed primarily to control of credit exposure and to collateral administration, rather than to computer operations per se. Looking forward, it seems likely that those guidelines will be further expanded and strengthened in light of the problems encountered in the government securities markets during 1985, and also that securities clearing operations will be among the areas covered in some detail as part of special "targeted examinations", which are part of the Federal Reserve's new supervisory initiatives adopted in October 1985.

In the last analysis, however, no mechanical system can be entirely fail-safe and also be commercially viable. Nor is it realistic to expect that even the most detailed bank examination procedures would or should detect every potential flaw in a bank's computer systems and related back-up arrangements. These limitations underscore the importance of retaining a close linkage between direct access to the operations of the payments system and access to the banking safety net, including the discount window, as well as acceptance by all participants in the payments mechanism of a regime of banking regulation and supervision.

Question 5. Please provide a listing of the dates and amounts of "daylight overdrafts" over $1 million incurred by U.S. banks since January 1, 1985.

Answer. See the enclosed tables, which give data on the frequency, size, and distribution of daylight overdrafts since early this year.

Question 6. In view of the comments of Federal Reserve Bank President Gerald Corrigan in his correspondence to Vice Chairman Preston Martin and your likely responses to question 2 above, I am concerned about the apparent concentration of responsibility for government securities transactions among the thirty-five or so "primary dealers". Are we not courting Continental-scale problems by permitting financial operations of such enormous magnitude and significance to be handled by so few institutions? Would there not be some risk-reducing, diversification benefits from limiting the responsibilities of any single institution and increasing the number of institutions involved?

Answer. The designation "primary dealer" applies to firms judged qualified to do business with the FRBNY, acting on behalf of the Open Market Account of the Federal Reserve System. Primary dealers must demonstrate financial strength, management depth and experience, and a commitment to participate in auctions and to make markets in a broad array of Treasury securities. Not surprisingly, in light of these qualifications, the primary dealers account for the bulk of dealer activity in Treasury securities. The condition of primary dealers is closely monitored through daily position and financing reports to the FRBNY, submission of regular financial statements and periodic on-site examinations of records and operations.

The potential number of primary dealers is not limited. Any firm that considers itself capable of meeting the criteria of the FRBNY may apply for primary dealer status. Its operations would be monitored for a time and its financial and managerial resources evaluated before such status were granted. Given the openness of this process, little would be gained by limiting the participation of any individual institution in the Treasury market, and any such restrictions could have a significant adverse effect on the liquidity of this key market sector.

In the context of the BONY problem, a related consideration is the degree of concentration of government securities clearing services, as distinct from securities trading. While the dozen or so large banks that are primary dealers typically clear their own securities, the approximately two dozen nonbank primary dealers, as well as numerous other nonbank dealers and brokers, rely upon banks to provide them with government securities clearing services. The market for those services is concentrated among five large clearing banks, of which one is BONY. The market structure which has evolved reflects economies of scale involved in offering highly automated large dollar payments services. Despite the relatively small number of such banks, the market for these services appears to be an intensely competitive one in terms of both price and nonprice service considerations.

Table 1
Daylight Overdrafts
Sum of Maximums at Each Institution
Daily Average Over Two-week Periods
(Millions of Dollars)

2-weeks Ending	Fedwire Funds Overdrafts	Fedwire Securities Overdrafts [1]	Total Overdrafts Subject to Control [2]
March 27	51,594	35,425	80,566
April 10	51,796	38,260	74,496
24	50,624	38,806	77,315
May 8	52,447	38,716	77,921
22	57,908	53,386	84,267
June 5	52,392	41,309	75,984
19	59,761	39,575	84,267
July 3	64,345	41,100	88,340
17	53,648	39,812	78,815
31	60,438	42,400	87,950
Aug. 14	54,772	40,640	83,731
28	53,346	43,163	80,221
Sept. 11	54,146	41,691	85,937
25	51,499	42,506	80,822
Oct. 9	54,208	39,923	84,674
23	55,048	41,843	83,764
Nov. 6	53,083	49,842	79,949
20	57,783	52,427	86,246
Dec. 4	53,444	56,708	80,889
18	52,056	52,307	83,168

Note: For each column, each institution's maximum net debit at any point during the day is selected, and these maximums are aggregated for each day and averaged over two weeks.

Division of dollar size data in Table 1 by number of institutions in Table 2 does not produce a meaningful average owing to the skewed distribution of daylight overdrafts; a relatively small number of larger banks account for a high proportion of the dollar volume of daylight overdrafts.

[1] The excess of book entry securities received over those sent.

[2] The total of funds overdrafts (column 1) less net securities credits (the excess of bank entry securities sent over book entry securities received) plus net debits on private wires.

Table 2
Number of Institutions in Daylight Overdraft
Daily Average Over Two-week Periods

2-weeks Ending		Fedwire		Total Overdrafts Subject to Control
		Funds Overdrafts	Securities Overdrafts	
March	27	1,517	260	1,598
April	10	1,458	260	1,536
	24	1,497	253	1,572
May	8	1,505	287	1,581
	22	1,494	275	1,564
June	5	1,420	261	1,485
	19	1,513	247	1,586
July	3	1,586	344	1,656
	17	1,328	383	1,383
	31	1,596	440	1,657
Aug.	14	1,412	439	1,475
	28	1,331	424	1,385
Sept.	11	1,460	507	1,516
	25	1,388	412	1,439
Oct.	9	1,369	407	1,423
	23	1,399	403	1,456
Nov.	6	1,354	443	1,406
	20	1,371	437	1,416
Dec.	4	1,200	425	1,245
	18	1,468	391	1,520

Note: Each column includes institutions in overdraft anytime during any day in two-week period.

Division of dollar size data in Table 1 by number of institutions in Table 2 does not produce a meaningful average owing to the skewed distribution of daylight overdrafts; a relatively small number of larger banks account for a high proportion of the dollar volume of daylight overdrafts.

Table 3a
Frequency Distribution of Institutions With Overdrafts During the Two-week Period Ending March 27, 1985

	Total No. of Institutions	Number of Days in Overdraft									
		1	2	3	4	5	6	7	8	9	10
-- Consolidated Assets --											
Less than or equal to $250 mil.	3176	1043	571	390	286	212	152	145	122	128	127
$250 mil. to $500 mil.	365	76	62	43	31	24	27	19	34	19	30
$500 mil to $1 bil.	235	49	33	18	29	21	15	16	8	23	23
$1 bil. to $5 bil.	297	47	32	28	19	26	26	16	22	31	50
$5 bil. to $10 bil.	62	4	3	2	2	1	4	1	10	8	27
Greater than $10 bil.	34	1	1	2	3	—	—	2	—	4	21
All	4169	1220	702	483	370	284	224	199	196	213	278

BLANK PAGE

BLANK PAGE

Table 3b
Frequency Distribution of Institutions With Overdrafts During the Two-week Period Ending December 18, 1985

	Total No. of Institutions	Number of Days in Overdraft									
		1	2	3	4	5	6	7	8	9	10
-- Consolidated Assets --											
Less than or equal to $250 mil.	3275	1069	625	425	295	251	155	131	125	105	94
$250 mil. to $500 mil.	374	100	75	44	37	21	30	23	9	22	13
$500 mil. to $1 bil.	230	62	36	32	21	18	15	7	10	9	20
$1 bil. to $5 bil.	321	68	56	33	30	19	17	13	16	23	46
$5 bil. to $10 bil.	61	5	8	5	1	5	2	2	3	7	23
Greater than $10 bil.	35	4	1	1	2	1	1	2	1	2	20
All	4296	1308	801	540	386	315	220	178	164	168	216

ASSIGNMENT AND SECURITY AGREEMENT: SECURITIES

This agreement (the "Agreement") is made as of ________ __, 198_ between ______________ (the "Borrower") and the FEDERAL RESERVE BANK OF NEW YORK (the "Bank").

1. As security for the payment of advances this date or hereafter made to Borrower by Bank under the continuing lending agreement between Bank and Borrower dated ______________________ (the "Continuing Lending Agreement"), and the Federal Reserve Act as implemented by Regulation A of the Board of Governors of the Federal Reserve System, and all other indebtedness, liabilities, and obligations of Borrower to Bank, including overdrafts or debit balances, now existing or hereafter incurred, matured or unmatured, direct or contingent, including extensions, renewals, changes in form, or checks or other items hereafter returned unpaid, together with interest thereon, Borrower hereby assigns and grants to Bank a security interest in the Collateral, together with the Borrower's right to the payment of money on account of the Collateral and with the rights of Borrower under any guarantee, assignment, pledge, or other agreement relating to the Collateral and all evidences of indebtedness, including notes and all insurance and other proceeds and accounts receivable arising therefrom (collectively referred to as the "Pledged Collateral"). "Collateral" means securities that satisfy one or more of the following: (1) Borrower has failed to

deliver against payment for its own account or for its client(s), (2) collateralize advances by Borrower to its client(s), or (3) Borrower has not yet received payment for from its client(s).

2. Except as disclosed to Bank in writing within one business day after the date of this Agreement, Borrower warrants that (a) it is the owner of or has the authority to assign the Pledged Collateral, (b) the Pledged Collateral is not subject to any existing or subsequent assignment, claim, lien, or other security interest of any character, or to attachment, levy, garnishment or other judicial process, or to claim for set-off, counterclaim, deduction or discount, and (c) with respect to the obligations of others, Borrower has not received notice of bankruptcy, insolvency, or adverse change in a debtor's, an issuer's, a drawer's, a guarantor's, or an acceptor's financial condition. Borrower will furnish to Bank at its request and certify as true and correct documents necessary to establish the foregoing warranties. Borrower also warrants that, after the Pledged Collateral is pledged to Bank, Borrower will neither perform nor fail to perform any act with respect to the Pledged Collateral that would impair Bank's security interest in the Pledged Collateral and will take all steps deemed necessary by Bank to perfect and maintain Bank's perfected security interest in the Pledged Collateral. Borrower will mark its records to show that the Pledged Collateral has been pledged to Bank and is subject exclusively to Bank's written orders and directions.

[illegible] AVAILABLE

3. Borrower further covenants, warrants and agrees, upon demand of Bank, immediately to deliver to Bank or to an agent or custodian designated by Bank all securities, assignments, bond powers, agreements, powers of attorney, guarantees, or other documents securing or in payment of or necessary for the transfer of Pledged Collateral, any endorsements or assignments to Bank being with recourse, and Borrower hereby waives presentment, demand, notice of dishonor, protest and notice of protest and all other notices relating thereto. Should the Bank exercise its rights as a secured creditor under the terms of the Continuing Lending Agreement, Federal law and the law of the State of New York, Borrower appoints Bank as its attorney in fact and authorizes Bank to conduct a sale of the Collateral, give such notices, and execute such documents as are necessary to convey good title to Collateral to any purchaser. This power of attorney is coupled with an interest and full power of substitution is granted to the assignee or holder.

4. If any provision of this Agreement shall be held invalid or unenforceable, such invalidity or unenforceability shall not affect any other provision hereof. The terms of this Agreement shall prevail over any inconsistent terms of the Continuing Lending Agreement, or any other agreement regarding advances between Bank and Borrower.

5. The Borrower hereby agrees to indemnify Bank and hold it harmless from and against any damages, liabilities, costs,

expenses (including attorneys' fees and expenses of litigation) or losses incurred by Bank or threatened against it which damages, liabilities, costs, expenses or losses shall arise due to a misrepresentation by the Borrower as to either (a) the right of the Borrower to pledge any Pledged Collateral pursuant to this Agreement, (b) the true owner of any Pledged Collateral, or (c) any claim of any person regarding the Collateral.

6. This Agreement and any other agreements regarding the Borrower's obligations to Bank are to be construed in accordance with, and governed by, the Federal law of the United States, and the law of the State of New York to the extent that the State law is not inconsistent with Federal law.

7. The warranties and indemnities of this Agreement shall survive its termination.

AGREED:	AGREED:
______________________________	FEDERAL RESERVE BANK OF NEW YORK
SIGNATURE: By: ______________	SIGNATURE: By: ______________
PRINT NAME: ________________	PRINT NAME: ________________
TITLE: ______________________	TITLE: ______________________

BEST COPY AVAILABLE

273

NASA Guidelines for Assuring the Adequacy and Appropriateness of Security Safeguards in Sensitive Applications

Chapter II - Software Development

1) A-71- Agency Head Responsibilities - Pg 2-1
2) Software Development Life Cycle - Pg 2-3
3) Risk Analysis - 2-17
4) Developing a Security Test Plan - Pg 2-35
5) Certification & Accreditation - Pg 2-37
6) Backup & Recovery - Pg 2-42
7) Security Test Procedures - Pg 2-46
8) Conducting Security Tests - Pg 2-63

Prepared for
National Aeronautics and Space Administration

MITRE

[redacted]

While it is common practice for systems developers to think of system functionality first and to delay security concerns until later, many opportunities to include effective controls are lost if not considered early [2]. To assure that system developers consider security throughout the software development life cycle (SDLC), OMB A-71, TM No. 1 requires the establishment of a management control process to assure that appropriate and adequate controls are incorporated into all applications.

2.1 [redacted]

Agency Head
Resp. for safeguards, etc

[redacted]

...In consideration of problems which have been identified in relation to existing practices, each agency's computer security program shall at a minimum: ...Establish a management control process to assure that appropriate administrative, physical and technical safeguards are incorporated into all new computer applications and significant modifications to existing computer applications. This control process should evaluate the sensitivity of each application. For sensitive applications, particularly those which process sensitive data or which

BEST COPY AVAILABLE

have a high potential for loss, such as automated decisionmaking systems, [redacted] The views and recommendations of the computer user organization, the computer installation and the individual responsible for security of the computer installation shall be sought and considered prior to the approval of the security specifications for the application. (2 [redacted] The objective of the design reviews should be to ascertain that the proposed design meets the approved security specifications. The objective of the system tests should be to verify that the planned administrative, physical and technical security requirements are operationally adequate prior to use of the system. The results of the design review and system test shall be fully documented and maintained as part of the official records of the agency. Upon completion of the system test, [redacted]

While the terminology used within OMB Circular A-71, TM No. 1 is not consistent with respect to the terms requirements, design and specifications, it is clear that the intent of the overall requirement for a management control process for the security of computer applications software is directed at ensuring that steps are taken to include security concerns and safeguards as an integral, albeit identifiable, part of the software development life cycle process.

2.2 The Software Development Life Cycle

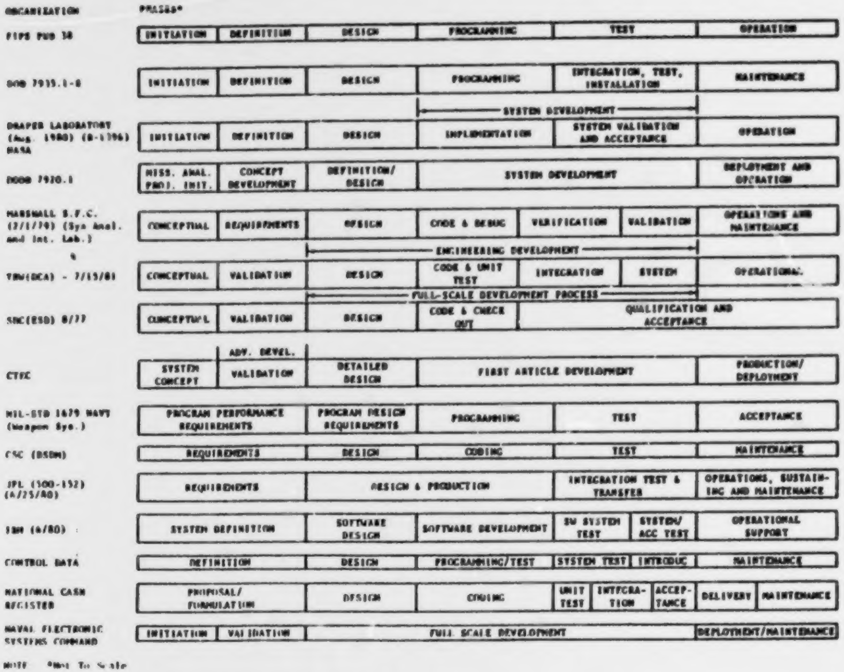
As noted previously, the software development life cycle (SDLC) is a technique used to divide the system development process into distinct phases. Figure 2-1 shows some 15 different software life cycle models. None of these models uses exactly the same terminology for all phases in the cycle. However it is important to note ...the management structure represented by the models is a proven method for enabling a project manager to: (1) estimate the cost/time to complete a system or software project; (2) make use of existing industry and government standards and guidelines; (3) assess the progress of a project at discrete points in the life cycle by conducting formal reviews and audits; and, (4) control system development by requiring go/no-go decision points throughout the life cycle [5].

Figure 2-2 presents a generic life cycle which is based upon the model presented in the National Bureau of Standards FIPS PUBs 38 and 64 [6,7]. The basic software life cycle, as depicted, identifies three major phases: initiation, development and operation. The development phase is divided into four stages: definition, design, programming and test. For the purpose of this document, the operations phase has been divided into two stages: implementation and maintenance.

2.2.1 The Initiation Phase

During the Initiation Phase, the objectives and general definitions of the requirements are established. First, there is an initial user definition activity. During this activity there is a determination of what's currently being done; what needs to be done; understanding the problem; defining of the

BEST COPY AVAILABLE



ORGANIZATION	PHASES*
FIPS PUB 38	INITIATION DEFINITION DESIGN PROGRAMMING TEST OPERATION
DOD 7935.1-S	INITIATION DEFINITION DESIGN PROGRAMMING INTEGRATION, TEST, INSTALLATION MAINTENANCE
	← SYSTEM DEVELOPMENT →
DRAPER LABORATORY (Aug. 1980) (R-1396) NASA	INITIATION DEFINITION DESIGN IMPLEMENTATION SYSTEM VALIDATION AND ACCEPTANCE OPERATION
DODD 7920.1	MISS. ANAL. PROJ. INIT. CONCEPT DEVELOPMENT DEFINITION/ DESIGN SYSTEM DEVELOPMENT DEPLOYMENT AND OPERATION
MARSHALL S.F.C. (2/1/79) (Sys Anal. and Int. Lab.)	CONCEPTUAL REQUIREMENTS DESIGN CODE & DEBUG VERIFICATION VALIDATION OPERATIONS AND MAINTENANCE
	← ENGINEERING DEVELOPMENT →
TRW(DCA) - 7/15/81	CONCEPTUAL VALIDATION DESIGN CODE & UNIT TEST INTEGRATION SYSTEM OPERATIONAL
	← FULL-SCALE DEVELOPMENT PROCESS →
SNC(ESD) 8/77	CONCEPTUAL VALIDATION DESIGN CODE & CHECK OUT QUALIFICATION AND ACCEPTANCE
	ADV. DEVEL.
CTEC	SYSTEM CONCEPT VALIDATION DETAILED DESIGN FIRST ARTICLE DEVELOPMENT PRODUCTION/ DEPLOYMENT
MIL-STD 1679 NAVY (Weapon Sys.)	PROGRAM PERFORMANCE REQUIREMENTS PROGRAM DESIGN REQUIREMENTS PROGRAMMING TEST ACCEPTANCE
CSC (DSDM)	REQUIREMENTS DESIGN CODING TEST MAINTENANCE
JPL (500-152) (6/25/80)	REQUIREMENTS DESIGN & PRODUCTION INTEGRATION TEST & TRANSFER OPERATIONS, SUSTAINING AND MAINTENANCE
IBM (6/80)	SYSTEM DEFINITION SOFTWARE DESIGN SOFTWARE DEVELOPMENT SW SYSTEM TEST SYSTEM/ ACC TEST OPERATIONAL SUPPORT
CONTROL DATA	DEFINITION DESIGN PROGRAMMING/TEST SYSTEM TEST INTRODUC MAINTENANCE
NATIONAL CASH REGISTER	PROPOSAL/ FORMULATION DESIGN CODING UNIT TEST INTEGRATION ACCEPTANCE DELIVERY MAINTENANCE
NAVAL ELECTRONIC SYSTEMS COMMAND	INITIATION VALIDATION FULL SCALE DEVELOPMENT DEPLOYMENT/MAINTENANCE

NOTE: *Not To Scale

FIGURE 2-1
SURVEY OF SOFTWARE LIFECYCLE MODELS

277

Initiation Phase	Development Phase				Operation Phase	
	Definition Stage	Design Stage	Programming Stage	Test Stage	Implementation Stage	Maintenance Stage
Present Request Document Feasibility Study Cost-Benefit Analysis	Functional Requirements Document Data Requirements Document	System Subsystem Specification Program Specification Data Base Specification	Users Manual Operations Manual Program Maintenance Manual			Change Requests
		Test Plan		Test Analysis Report		

(Source: FIPS PUB 38)

FIGURE 2-2
THE SOFTWARE LIFECYCLE

scope, objectives and operating environment; definition of functional, performance, and methodological requirements; and, acceptance criteria.

The second activity conducted during the Initiation Phase is evaluation and initiation of necessary documents to formally commence the software development project. This activity includes performing a comprehensive study of technical, operational and economic feasibility; performance of a cost/benefit analysis; analysis of general design approaches, and generating a development plan.

Documentation produced during the Initiation Phase requires user involvement to define the project and its worth. Typically, a Project Request Document is developed as a means for the user organization to request the development, procurement or modification of software or other ADP-related services. It serves as the initiating document in the software life cycle, and provides a basis for communicating with the requesting organization to further analyze requirements and assess impacts. The second document produced is usually the Feasibility Study Document. The purpose of this document is to provide: (1) an analysis of the objectives, requirements and system concepts; (2) an evaluation of alternative approaches for reasonably achieving the objectives; and (3) identification of a proposed approach. The third document, Cost/Benefit Analysis, provides managers, users, designers and auditors with adequate cost and benefit information to analyze and evaluate alternative approaches.

All documentation is widely reviewed and is followed by a management decision of whether to continue on to the

Definition Stage. For external procurements, a Request for Proposal (RFP) is issued, proposals are evaluated and a contract is awarded.

2.2.2 The Development Phase

During the Development Phase, the requirements for software are determined and the software is then defined, specified, programmed and tested. The Development Phase is broken down into four stages: Definition, Design, Programming and Test.

2.2.2.1 The Definition Stage

The activities during the Definition Stage include: translation of the user requirements into detailed function requirements and a functional architecture defining the operating environment, functional modules, inputs, outputs, processing requirements and system performance requirements (as needed to meet user performance requirements); definition of data requirements; completion of a general top-level design; definition of functional interfaces (man/machine, system/system, function/function); identification of required equipment; and, planning for development activities. Documents typically produced during the Definition Stage include the Functional Requirements Document and the Data Requirements Document.

2.2.2.2 The Design Stage

During the Design Stage, the Systems and Program Specifications are developed. Activities at this point include: designing the system to meet functional requirements; dividing functional modules into program modules identifying the inputs, processing and outputs of each; definition of the control and data structures and protocols; and, specification of interfaces in detail. Documents typically produced during this stage include the System/Subsystem Specifications, Program Specifications, Data Base Specifications and the Test Plan.

2.2.2.3 The Programming Stage

During the Programming Stage, the software is coded and debugged. Activities may include obtaining of the required hardware; writing, testing and debugging of software programs; preparation of manuals; and, completion of Test Procedures. Documentation typically produced during the Programming Stage includes the Users Manual, Operations Manual, Program Maintenance Manual and the Test Plan.

2.2.2.4 The Test Stage

During the Test Stage the software is tested and the related documentation is reviewed. The software and documentation are evaluated in terms of readiness for implementation. Activities include: performance of integration and acceptance testing; training of users and operators; installation in the operational environment; data base conversion; and testing in the operational environment. Documentation produced during this stage is the Test Analysis Report.

2.2.3 The Operations Phase

During the Operations Phase, software is maintained and enhanced as additional requirements are identified. The Operations Phase can be viewed as two distinct stages: Implementation and Maintenance.

2.2.3.1 Implementation Stage.

The Implementation Stage is that period immediately following the beginning of the operation of the software in a production mode. Activities during this stage include the removal of latent defects that were not discovered during development testing. Additionally, at some point in the later period of the programming or test stages when no additional user requirements are permitted and the baseline is frozen to permit testing. During this implementation stage, the user requirements that were generated after the baseline was frozen, are incorporated as modifications to the operational software.

2.2.3.2 The Maintenance Stage.

Following the initial implementation stage on-going maintenance is required. Maintenance activities over the remaining life time of the application can be broken down into three distinct types [9]:

- Corrective maintenance-fixing errors that had not been uncovered or fixed correctly if previously exposed in testing.
- Adaptive maintenance--performed in anticipation of hardware or support software changes.
- Perfective maintenance--changes to improve performance.

2.3 Software Development Life Cycle Security Activities

Much of what needs to be done to improve (or initially incorporate) security is not clearly separable from what is needed to improve the usefulness, reliability, effectiveness, and efficiency of computer applications [2]. While security concerns should be integrated in the life cycle of a computer application, the steps taken to ensure the appropriateness, adequacy and reasonableness of security safeguards should be separately identifiable activities within each stage or phase of the SDLC.

2.3.1 Security Activities

System planners, developers and users should accomplish a series of security-related actions throughout the SDLC. While the order in which the actions should be accomplished is presented sequentially, it should be recognized that there will be much interaction between a particular step and the preceding steps. The process for incorporating security safeguards in an application is not substantially different from the SDLC activities identified in Section 2.2. It should also be noted that if during development, any change occurs in a software requirement or specification, the change must be reviewed to determine if coincidental changes are required in the security requirements or specifications.

The security activities (Figure 2-3) to be completed throughout the SDLC are:

- Determine Sensitivity of Data/Application
- Determine the Security Objective(s)
- Assess the Security Risks
- Conduct a Security Feasibility Study
- Define the Security Requirements
- Develop the Security Test Plan

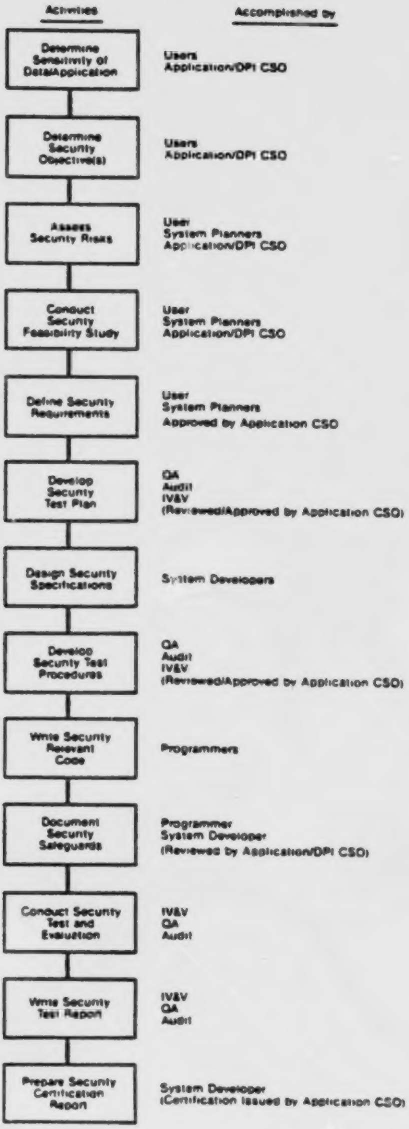

Activities
Accomplished by
Determine Sensitivity of Data/Application
Users
Application/DPI CSO
Determine Security Objective(s)
Users
Application/DPI CSO
Assess Security Risks
User
System Planners
Application/DPI CSO
Conduct Security Feasibility Study
User
System Planners
Application/DPI CSO
Define Security Requirements
User
System Planners
Approved by Application CSO
Develop Security Test Plan
QA
Audit
IV&V
(Reviewed/Approved by Application CSO)
Design Security Specifications
System Developers
Develop Security Test Procedures
QA
Audit
IV&V
(Reviewed/Approved by Application CSO)
Write Security Relevant Code
Programmers
Document Security Safeguards
Programmer
System Developer
(Reviewed by Application/DPI CSO)
Conduct Security Test and Evaluation
IV&V
QA
Audit
Write Security Test Report
IV&V
QA
Audit
Prepare Security Certification Report
System Developer
(Certification Issued by Application CSO)

FIGURE 2-3
SOFTWARE DEVELOPMENT LIFECYCLE SECURITY ACTIVITIES

[redacted]

Much of the information provided in the description of the security activities is presented in FIPS PUB 73 [2]. The information is incorporated herein to provide the reader with a complete discussion.

2.3.1.1 Determine the Sensitivity of Data/Application

The degree of sensitivity of an application system depends upon the data it will process and/or the types of functions to be accomplished by the software. For example, data may be personal in nature, represent valuable tangible assets such as high-dollar value inventory or represent real dollars. Application processes which perform critical operations may include formulas or algorithms that must always be executed exactly the same, such as engineering calculations or on-board software for a space vehicle.

FIPS PUB 73, Section 2.3, provides categories of sensitive systems with some examples of the types of applications that would fall under each of the categories. The categories and examples are:

- Applications Providing General Processing Support - The primary concern is for accidents, errors and ommissions. Health, safety, welfare and lives may depend on the correctness of output from these applications. Thus data integrity, including integrity of the software, is critical.

- Engineering calculations used in aircraft design
- Query systems that support health care decisionmaking
- Automated wind tunnel control systems
- Simulation of the dispersion of toxic substances

- Funds Disbursement, Accounting, Asset Management Systems - These systems frequently involve personal or other confidential data. In these applications, deliberate and accidental acts are a major concern. Data integrity is the major objective. Data confidentiality may also be required.
  - Payroll
  - Financial accounting
  - Procurement support
  - Equipment inventory control

- General-Purpose Information Systems - The simultaneous use by different user populations makes data confidentiality critical as well as data integrity. The generality of such systems and their associated security requirements also make it more difficult to design effective security controls.
  - Generalized data management systems
  - Centralized management information systems

- Automated Decisionmaking Systems - Manual review is more difficult, so that errors made by the automated systems are less likely to be detected before they lead to serious harm. The major objective is rigorous data integrity.
  - Fully automated funds disbursement and accounting systems
  - Automated inventory reordering
  - Automated scheduling for maintenance

- Real-Time Control Systems - These systems have all the security concerns of automated decisionmaking systems plus a rigorous requirement for constant availability and very rapid response times. Basic controls plus automated fault detection, backup, and recovery in conjunction with redundant hardware support may be required.
    - Air traffic control
    - NASA mission control
    - Rapid transit system control
    - Automated production control

- Systems Affecting National Security or Well-Being - These systems must be protected against hostile acts by unauthorized persons who have considerable resources. Data integrity, confidentiality, and ADP availability are all required plus techniques for formal development and verification of controls in operating systems as well as application systems.
    - Military command and control
    - Management of multilevel classified information
    - Integrated electronic funds transfer
    - Nuclear material control and accountability

Additional guidance on determining the sensitivity of the data/application can be found in NASA Handbook (NHB) 2410.1 [10]. For the purposes of the NASA Computer Security Program, a sensitive application is defined as the use of a computer system for processing classified, proprietary, dollar sensitive, time sensitive, or Privacy Act data. All other computer system use, such as that for scientific, technical, research, or development activity, may be considered as a nonsensitive application; however, this does not preclude such uses as being designated a sensitive application if this will provide necessary and useful controls. Designation of a test and mission control application as a sensitive application is within the prerogative of responsible personnel. For test and mission control applications, it is recommended that security measures be provided for as an element of mission or test plans. This will allow any necessary security measures to be tailored to specific test and mission needs in a manner that provides a sound balance between requirements for controls and for operational flexibility.

To assist in determining whether an application may be sensitive, NHB 2410.1 provides the following guidance. First, careful exercise of judgement is required in evaluating the sensitivity of applications. In those instances where it is not clear that an application is sensitive, it is necessary to weigh the intangible costs of potential loss against the cost to protect the application if it is categorized as sensitive. This type of cost-benefit analysis is especially critical in evaluating research, development, test, and mission control applications. The possibility that applications are sensitive only under certain conditions should not be overlooked. For example, it could be useful to categorize unique mission control applications as being sensitive only during a specific period of the mission.

Essentially, the objective of this step is to determine whether or not the application or the data is sensitive. If the data or application is sensitive, the rationale should be documented. When reviewing the data, one should attempt to determine the potential gain to persons from unauthorized access to or use of the data or the application process. This step should be accomplished by the owner or intended user in concert with the application and/or DPI CSO.

If the application is determined to be sensitive, the next step in the process is to determine the security objective or objectives.

2.3.1.2 Determine The Security Objective(s)

There are two types of events that can have unwanted or undesirable effects on sensitive data or applications; advertent (deliberate) or inadvertent (accidental). The advertent or inadvertent events may result in the modification, destruction, or disclosure of data or applications software programs, or the unavailability of computing resources. A useful approach for assuring that appropriate and adequate safeguards are incorporated in sensitive applications is to establish security objectives that, if achieved, will reasonably mitigate advertent or inadvertent events. Generally speaking, five security objectives should satisfy all types of events and effects: data integrity, application integrity, data confidentiality, application confidentiality and ADP availability.

- Data Integrity - The state that exists when computerized data is the same as that in the source documents or has been correctly computed from source data and has not been exposed to accidental or malicious alteration or destruction [2].
- Applications Integrity - The state that exists when the source and object code are the same as originally developed and certified/accredited or, have been modified and tested in accordance with established standards and procedures and recertified/reaccredited, and have not been exposed to accidental or malicious alteration or destruction.
- Data Confidentiality - The state that exists when data is held in confidence and is protected from unauthorized disclosure [2].

- Application Confidentiality - The state that exists when application source and object code and documentation is held in confidence and is protected from unauthorized disclosure.

- ADP Availability - The state that exists when required ADP services can be obtained within an acceptable period of time [2].

Figure 2-4 provides a guide for determining sensitive application security objectives. First, determine the category of sensitive application. Second, refer to Figure 2-4 and determine the possible security objective that may have to be achieved. It should be noted that this is a preliminary determination. The assessment of risks and security feasibility study may surface concerns or limitations that would require a modification of the security objectives.

This step should be accomplished by the owner or intended user in concert with the application and/or DPI CSO.

2.3.1.3 Assess the Security Risks

The types of controls that will ultimately be incorporated into an application system should be determined based upon the potential loss or harm that could be suffered if the data or the application were modified, destroyed or disclosed or is caused to become unavailable due unauthorized or undesirable events. FIPS PUB 102 [8] provides an introductory discussion of the usefulness of risk analysis for assessing the risks to a new application.

> The primary purpose of risk analysis is to understand the security problem by identifying security risks, determining their magnitude, and identifying areas where safeguards or controls are needed. It can also

Effect	Type of Event	Category of Sensitive Application					
		General Processing Support	Funds Disb Accounting Asset Mgmt	Gen. Purpose Information Systems	Automated Decisionmaking Systems	Real-Time Control	National Security or Well-Being
Modification of Data/Appl	Advertent	AI	DI, AI	DI, AI	AI	AI, DI	AI, DI
	Inadvertent	DI, AI	AI, DI		DI	AI, DI	AI, DI
Destruction of Data/Appl	Advertent			DI, AI	AI	AI, DI	AI, DI
	Inadvertent			DI	DI	AI, DI	AI, DI
Disclosure of Data	Advertent		DC	DC	AC	AC	DC, AC
	Inadvertent			DC		AC	DC
Unavailability of Computing Resources	Advertent		AA	AA		AA	AA
	Inadvertent	AA			AA	AA	AA

Legend:
DI = Data Integrity
DC = Data Confidentiality
AA = ADP Availability
AI = Application Integrity
AC = Application Confidentiality

FIGURE 2-4
SENSITIVE APPLICATION SECURITY OBJECTIVES

be used to determine how many resources to budget for security and, with user inputs and policy requirements, can provide the basis for choosing system security requirements.

Risk analysis can also be useful in validating requirements. If requirements are defined to the functional safeguards level, risk analysis can be used to determine whether the protection embodied in the controls reduces expected loss to an acceptable level at acceptable cost.

In the initial assessment of risk, the concern is for the impact and frequency of major failures. The impact of major failures can be described in any convenient terms--dollar value of loss, extent of inconvenience or hardship, lives lost or degree of disruption to the national security or agency mission.

The impact of at least the following failures should be assessed for each major body of information that is to be processed by the proposed system [2].

- Inaccurate Data - Data (programs) could be corrupted with errors, but the system continues to function while producing erroneous outputs. Estimate the potential impact of erroneous actions that might result assuming only that the output of the ADP system is not so obviously out of line with reality that the errors would be noticed. Consider both the impact of a few very serious errors and the cumulative effect of many small errors.

- Falsified Data - An individual could falsify data in order to gain some advantage. The falsifications may be limited only by the fact that they are subtle enough so they are not detected manually. Estimate the total impact that could occur over an extended period of time.

- Disclosed Data - Sensitive data in the system becomes available publicly or to certain individuals. The unauthorized disclosure of data is not necessarily discovered.

> **Lost Data or Application Software Code or Documentation** - Data, source code, object code or documentation are destroyed or corrupted. Backup versions are nonexistent or not usable, and the data must be reconstructed manually or software code or documentation must be rewritten. Estimate the impact of losing the data, source code, object code or documentation. If manual reconstruction is obviously not feasible and if backup in depth is anticipated, estimate the impact of using old version, inaccuracies, and the temporary unavailability that would result while recovering from an old copy on the assumption that all current or recent backup copies have been destroyed.
>
> **Unavailable Data Services** - Estimate the impact if the computer hardware or related equipment in the computer facility (DPI) is disabled and the system is not available until it can be brought up in another facility.

An estimate of the impact of a major security failure is not particularly meaningful without some estimate of how frequently it might occur. Unfortunately, during the initial planning for an application system, it is difficult, if not impossible, to estimate the frequency of a major failure by evaluating the controls in place. However, it is possible to develop rough estimates based upon experience with manual system activities and by looking at the experience of failures or disruptions with similar types of systems. FIPS PUB 73 [2] provides the following guidance:

> If the proposed system is generally comparable to other computer applications, then a major security failure of the sort described above can be estimated to occur once in a hundred years. This simple estimate is based on the following line of reasoning:
>
> > Available security controls (if they are properly managed) can prevent major security failures from reoccurring as frequently as once every 10 years. On the other hand, any ADP system has several

> vulnerabilities against which there is little defense; most systems can be manipulated by any one of several individuals who are in a position of trust—programmers, those responsible for security, the computer operators, and others. There are enough instances of major security failures in computer applications so that an expected frequency of once in a thousand years is very optimistic. The estimate of once in a hundred years is only intended to be accurate within an order of magnitude.

An alternate approach for estimating frequency is to use a low, medium or high frequency rather than orders of magnitude. This approach when used as part of the Threat Team Analysis should provide sufficient data to provide an assessment of the impact of major failures.

A threat team is composed of key employees within an organization who meet as a task force to search for threats and vulnerabilities in a system and who create possible scenarios for attacking the system. Use of such teams is based on the premise that people in the best position to discover how to beat the system are those who work with it every day. The objective is to capitalize on their knowledge. Threat analysis unlocks this potential through the use of a moderator familiar with computer abuse methods. The threat sessions seek a symbiosis between the moderator's general knowledge of typical schemes and the participants' specific knowledge of data processing operations and functional experience with the system under evaluation [1]. The threat team/analysis approach has been used successfully in NASA in the evaluation/certification of existing sensitive systems [12].

Actual threat scenario development is accomplished through a series, usually two, informal team meetings. The first meeting should be scheduled for two to two and one-half hours. The second meeting should last for one to two hours. At the beginning of the first meeting of the team, a basic set of ground rules and background information must be given. The key elements of this background are summarized in Figure 2-5 [11]. The moderator should give a brief summary of the statistics of computer abuse, an outline of major schemes that have been perpetrated against similar types of applications and how they were accomplished (modus operandi). The objective is to "prime the pump" and stimulate the participants in developing possible attacks against their system.

Flip charts should be used to record the following information:

- What is being attacked or compromised in the system?
- What vulnerabilities would permit the attack to be accomplished?
- What methods or procedures would be utilized?
- What types of safeguards or controls could be used to prevent or reduce loss?
- What is the likelihood that this scenario will work (high, medium, low)?
- What is the impact on the system or organization if the scenario were successfully executed (order of magnitude dollars, delays, etc.)? Note: Quantification of impact while desirable is not mandatory.

One alternate approach is for each threat team participant to keep notes, summarize each scenario and turn the notes over to the moderator for summarizing. The summary would be reviewed by each team member at the second meeting. Another alternative, is for the moderator, or other designated person,

1. Introduction	The moderator is introduced by a NASA management official. Participants have not been informed in advance of the subject of the meeting. The official then leaves. The meeting place should be around a large table in a comfortable room, such as the board room (thus giving status and approval to the project).
2. Summary	The objectives of the study are explained; the time frame and the responsibilities of the participants are outlined.
3. Moderator's Identification	The moderator should establish his position within the group and define his own role, which is that of a resource person. He knows a good deal about computer abuse schemes in general, but little about how this particular organization operates.
4. Why this Organization?	Participants should be put at ease by explaining that the study is simply a precautionary excerise; there is no reason to suspect an on-going perpetration.
5. Why these Participants?	Participants often wonder "why me" at this stage. Again they must be reassured. They have been selected for the study because of their knowledge and experience; they are the people who "really know how this business works."
6. Warning about Secrecy	Participants are asked not to discuss the subject or content of the meetings outside the group.
7. Schedule	The group will meet for several hours. A transcript will be prepared and circulated for changes. A second meeting of the same group is scheduled in about one week, and a final report circulated in the same fashion.

FIGURE 2-5
OUTLINE OF INTRODUCTORY COMMENTS
BY MODERATOR AT FIRST THREAT SESSION

to act as a recorder, to keep a "transcript" of the meeting. The transcript would then be reviewed by each participant after each meeting and finalized as a consensus record.

The second meeting should be conducted within one to two weeks after the initial session. The initial session provides a "sensitizing" of the participants. Between the two sessions, participants can review the proposed system with some new perspectives and will usually provide additional scenarios. At the conclusion of the second meeting, conclusions about the scenarios should be agreed upon by all participants. A report of the sessions should be written. The report should be closely held, distribution extremely limited and copies protected.

Upon completion of the threat scenario analysis exercise, it is suggested that the security objectives be reviewed to determine if the preliminary security objectives are still valid. When reviewing the security objectives, the following items should be considered to ensure that major concerns about security have been addressed [2].

- Source data accuracy - Will the data supplied to the ADP system be accurate and complete enough to support its intended uses without harmful side effects?
- User identity verification - Can users of the systems be adequately identified and authenticated so they can be held accountable for their actions?
- Restricted interfaces - Are the user interfaces to the system sufficiently restricted so that adequate security is feasible?
- Separation of duties - Do the boundaries between ADP and related manual activities provide maximum separation of duties and independent review?

- Facility security - Is the proposed processing facility adequately secure?

The next step in the security life cycle is to determine the types of controls that should be incorporated into the application that will achieve the security objectives.

2.3.1.4 Security Feasibility Study

The purpose of the security feasibility study is to determine if controls are available to meet the security objectives, how well they will satisfy the objectives, whether the controls should be preventive, detective or recuperative in nature and what mix of administrative, physical and technical controls is most feasible. Cost, performance degradation, and impact on requirements for user friendliness should be considered. In other words, what types of controls are appropriate for the proposed application.

A key to the feasibility study is the use of an appropriate methodology to analyze the proposed application to determine what security controls are available and how well they meet the security objectives. Brill, in "Building Controls Into Structured Systems" [13], divides applicatin controls into three major classes: controls over inputs, controls over processing and controls over output. Brill's methodology is based on a hierarchical approach that leads the user through a tree structure to address a variety of control issues such as input authorization, internal data movement, operator intervention, and output distribution.

At this point in the life cycle with the determination of sensitivity completed, the identification of security objectives, the assessement of risks and security feasibility completed, definition of security requirements is the next step.

2.3.1.5 Define Security Requirements

Definition of security requirements takes place during the definition stage of the development phase of the system development life cycle. The term "requirements" can be used at many different levels. The requirements defined at this point should include everything that the users and other responsible parties want to require of the application software. The security requirements should be expressed in a way that permits the software designers to choose the best way of implementing them. It should be remembered that security controls can be enforced either by software or by physical or administrative procedures. For example, data integrity can be checked by human review or by automated bounds and consistency checks. When possible, it is recommended that controls be implemented in software for the following reasons: once controls have been automated, the continuing cost to enforce the controls is usually lower than when enforced manually; and, automated controls will be applied more consistently.

It should be noted that FIPS PUB 73 [2] indicates that the documentation of security requirements constitutes the security specification called for in OMB Circular A-71, Transmittal Memorandum No. 1, paragraph 4.c (1). Also, security specifications may be incorporated into the functional requirements document and the data requirements document as called for in FIPS PUB 38 [6], or it may be an independent document.

The first step in defining security requirements is to conduct in analysis of the "current system" to identify and develop an understanding of the principle functions and to identify sources of input and the flow of data through the system. When the current system is reasonably understood, the user and system requirements should be documented. The sources of input and the flow of data through the system are two of the most important sources of data for defining security requirements. Brill [13] indicates that systems analysts can use four different sources to identify controls that belong in the new system: stated user policy, unstated user policy, the current system and external constraints.

- Stated user policy--The best way to begin to determine the controls needed in a system is to ask the users about the need for controls. Users have a genuine stake in the new system as well as thorough knowledge of their own requirements. Users should be asked how they handle errors in the present system, the kinds of errors or problems they suspect they don't know about (that is, those that are slipping through undetected). Users should also be asked about laws or regulations that affect the way system processes must be done as well as questions about the value and criticality of their data.

- Unstated user policy--Users expect comprehensive controls to be built into their system without ever mentioning controls in discussions with the analysis team. For example, the need to test check digits on account numbers may never be stated because the present manual system may not have facilities to permit it. But users may assume that an automated system tests check digits as a matter of course and that you don't need to be told to build such tests into the system. Systems planners must learn the system--and understand the problem--well enough to challenge the unstated assumptions of the users and turn unstated requirements into stated ones.

- The current system--New systems tend to do many of the same things as the systems they replace. Of course, they may do them differently (via a terminal, for example, rather than through batch-produced reports). So, while the specific controls in the new system may differ from those in the old system, there are going to be overlaps. But, you have to look for them, and have to recognize them as controls and to assess their suitability for transplantation into the new system.

- External constraints--Various laws and regulations can directly affect a system. (This area would include agency policy for security, financial accounting, etc.)

FIPS PUB 73 [2] suggests that the following areas be addressed when developing security requirements: identification and definition of systems interfaces (to include responsibilities associated with each interface and a separation of duties), identification of the sensitive objects to be processed, determination of error tolerances and definition of availability requirements.

- Identification and definition of systems interfaces--Identify each job function which is related to the application system. Consider each job function as an interface to the application. Define the nature of the interaction between each job function and the system. Also identify and define any interfaces to other automated systems. Include all job functions (or other automated systems) that are to be supported by this system--even if the people performing those jobs only receive reports from it. Also include all job functions (or other automated systems) that supply information to the application system or that support its operation. Be sure to include critical job functions such as: source data collection, input preparation, data entry, output dissemination, data base administration, system security planning and control, internal audit, application program maintenance, archival or backup data storage, computer operations, and system programming.

Define the responsibilities of the individuals who interact with the application system through each defined interface. Identify the constraints on the use of each interface that must be enforced if security is to be preserved. Consider the likelihood of errors occurring in the use of the interface and identify the consequences. Consider the consequences of deliberate misuse of the interface. Identify the management and administrative controls that will be available to ensure that the interface is used properly.

Examine interfaces to ensure that security exposure will be minimized even if an interface is misused. Ideally, any action that could result in serious harm should be checked or approved from an independent interface.

- Identification of sensitive objects to be processed--Identify the objects to be processed--include input data, stored data and output data. Determine the sensitivity and asset value of the data objects. Identify the operations or functions that users will perform on this data.

- [redacted] Real-time control systems such as air traffic control have virtually no margin for error since human lives may be lost. Some management information systems, particularly those used to predict future demands and resource requirements, may not be as susceptible to errors in data. However, algorithms in the code may have less tolerance for errors. The application's tolerance for error and the requirements for maintaining error levels within acceptable tolerance must be defined.

- Definition of availability requirements—Determine the user tolerance for interruption of output data and the potential harm that could be a result due to non-availability of the application output.

The preceding discussion in this section has addressed the collection and analysis of data that precedes the actual documentation of security requirements. Requirements should define what is required by the user not how it is to be accomplished.

FIPS PUB 73 [2] describes some basic controls that can be used to achieve security objectives. It is relevant at this point in this discussion to summarize these basic controls since the descriptions provided, with some modification, can be used as security requirements. The basic controls provided in FIPS PUB 73 are: data validation, user identity verification, authorization, journaling, variance detection, and encryption.

- Data validation—Invalid data may lead to erroneous outputs, can destroy the credibility of the system, demoralize those trying to use it, cause excessive system maintenance costs, and, in extreme cases, cause the system to become unavailable or unusable.

  Data validation involves the examination of computerized data to determine if it is accurate, complete, consistent, unambiguous, and reasonable. Direct evaluation methods (discussed below) are not able to find all errors. Data integrity depends on the correctness and integrity of all the activities by which the data is collected and processed. Data validation is a very basic control, but it should only be expected to detect gross errors and it will not compensate for poor control over other aspects of the application system. Data should be validated

> during data collection and entry--prior to its use by the system; and, continuously, as new data is generated or used during processing.
>
> Data validation should be required during data entry and during processing. Automated editing and validation should be used in both batch and on-line systems. In batch processing systems, validation routines may run against input data before it is processed. Alternately, validation can occur as each transaction is processed. Transactions that contain errors should be recorded on a file for correction at a later time. On-line systems can provide the data entry personnel with immediate validation information so detected errors can be corrected immediately. During the definition stage the editing and validation technique to be employed is not specified, rather the requirement should state that all data originating from hard copy should be validated prior to the transaction being entered into the system.

For a batch system a typical requirement statement might read:

> All source data will be keyed twice and automatically compared with the transcribed source data previously keyed.

For an on-line system where transactions are entered in real-time, a typical security requirement might read:

> All keyed transactions (or transactions of a certain type) will be visually verified prior to transmission to the system.

Data may also be validated during processing. Most of the techniques that are appropriate to validation during data entry may be applied during processing. An example of a requirement that related to validation during processing is:

Transaction with errors detected during the data processing phase need to be controlled to ensure they are corrected and reentered in a timely phase.

- User identity verification—Identification occurs when the user provides an identifier-the name by which the user is known to the system. The user's identifier is unique, unlikely to change and need not be kept secret. It is used during processing for authorization controls, variance detection and for other purpose such as accounting and billing. Verification occurs when the individual passes some further test which "proves" that the user is actually the person associated with the identifier. This is also called user authentication.

Examples of requirements for user identification and/or verification might be as follows:

For batch submissions, users must be visually identified by a control clerk and all jobs logged.

For on-line submission of transactions, all users must have an individual identifier and password for initial logging on.

- Authorization—Once a user's identity has been verified, the application may still need to check each request for service to determine whether it is a legitimate request by that user. Some users may be authorized to perform some functions but not others and to have access to perform some functions but not others and to have access to some data but not to other data. In some cases, the authorization decision may depend on not only WHO is requesting what MODE OF ACCESS to which OBJECT, but also on other easily testable conditions. The time of day, the day of the week, previously detected security variances, or other concurrent activity might be used to affect the authorization decision.

An example of this type of authorization when stated as a security requirement might read:

The system must be able to restrict update access to specific time of the day and days of the week.

- Journaling--Journals may be employed to log activities or events external to the operational environment or those internal to the application system. Journals of external events in the operational environment can be maintained manually while journals used to record activities internal to the application must be automated. From a security standpoint, the ideal journal would include a 100 percent recording of all events relating to data, software, and system resources. From a practical standpoint, such a journal may, in some systems, be out of the question since the overhead to record all events would reduce system response to less than acceptable levels of performance. Requirements for journaling should be carefully considered, reduced to formal statements and be stated in positive terms. Items that should be included are:

  - definitions of what kind of data is to be protected and how the system will recognize such data,

  - the degree of accuracy that is necessary for various types of data, and

  - the definition of who is authorized to access protected data and how the system would recognize an authorized user.

Examples of a security requirement for journaling are:

The system will log all accesses by payroll personnel to any employee's payroll record. The system will log all initial log-ons, final log-off at the end of the normal work day and all log-ons and log-offs on weekends.

The system will log all opening and closings of the payroll master file and payroll transaction update files.

- Variance detection—The objective of variance detection is to allow management to detect and react to departures from established rules and procedures that it has determined may constitute hazards. Variance detection acts as a strong deterrent to authorized users abusing their privileges since they perceive the risk of detection to be unacceptably high. Variance detection is useful whenever it is not practical to prevent the undesirable activity by means of an authorization mechanism. In some cases, there may be no way to determine in advance whether an action should be prevented. The mechanisms required to support variance detection are related to mechanisms needed for other purposes. Recovery, accounting, load-balancing, tuning and the identification of recurring user difficulties all require some of the same capabilities.

A security requirements for variance detection might read:

> The system must be capable of providing post-processing analysis of all or selected activity initiated from a given terminal or by a given employee.
>
> The system must provide an interaction capability to identify attempted accesses to restricted files by unauthorized users.

The preceding are examples of static monitoring. Some variances can and should be detected in real-time so that responses can be immediate. An example of a dynamic monitoring requirement might read:

> The system must be capable of real-time display at a designated console of the full interactive traffic of any terminal or user.

- Encryption—The applications that are most likely to need encryption are those that transmit highly confidential data across communication lines. Applications that transmit financial transactions or other critical data may also need encryption if some is likely to derive enough benefit from modifying the data during transmission to compensate for the risk and cost of the effort. Encryption of data in storage is an alternative that may be more cost effective than other storage security controls—especially when appropriate support for encryption is readily available.

A security requirement for encryption might read:

> All data transmitted between the host computer and remote site will be protected from unauthorized disclosure and modification during transmission.

Following the definition of security requirements, two activities should be initiated: development of the test plan and the design of the security specifications. While they are shown in Figure 2-3 as sequential events, they can be accomplished in parallel since two different groups of people are involved.

2.3.1.6 Develop the Security Test Plan

Testing and evaluation attempts to demonstrate that a system is reliable, meets specifications, and meets the requirements of the user. Sorkowitz [14] provides the following comments on testing:

> The main problem in program testing becomes clear when we try to define the word "testing." To many programmers, testing is a process of proving that a program is correct (i.e., the program performs according to specification). However, experience leads us to the belief that there really is no practical way to demonstrate that a program is correct. The best we can say is that at some point in time, there are no known errors. Myers [4] gives a different definition of testing: "Testing is the process of executing a program with the intention of finding errors."

Careful and thorough testing and evaluation can improve system security by uncovering errors, omissions and other flaws in the system's design and code [2]. From a security perspective, the testing of security controls should focus on ensuring that

[redacted]

[redacted]hat is to be tested, the testing schedule, resource requirements, testing materials, requirements for test training, the location of the test, the functional security requirements, the tests to be performed on the software and their relationship to the functional security requirements, the testing methodology, the evaluation criteria, data reduction techniques and a description of each test to be performed. The test plan format provided in FIPS PUB 38 [6], with slight modification, can be used to develop the security test plan. A suggested test plan format is provided in Appendix A.

At this point in the system development life cycle, only portions of the test plan can be developed. If the format at Appendix A is utilized, the following sections can be generated at this time. Section 1, General Information, can be completed. In Section 2, Plan, the software description can be written; tentative milestones can be developed; and, preliminary development of the testing subsection can be started. One of the critical items in Section 2 is the area of test training. Specifically, the types of training for the test team should be identified as soon as possible. In Section 3, under Specifications and Evaluation, the functional security requirements and the security functions to be tested can be identified. Additional portions of the test plan will not be able to be developed until such time as the system/subsystem specifications, program specifications, data base specifications and security specifications have been generated.

One of the major objectives of the security test and evaluation of an application under development is to provide some of the data to support the certification of the security controls as required by OMB Circular A-71, Transmittal Memorandum No. 1. FIPS PUB 102, Guideline for Computer Security Certification and Accreditation [8], describes how to establish and carry out a certification and accreditation program for computer security. FIPS PUB 102 also provides some guidance that is useful in developing security test plans. It should be noted that in the NASA environment a different set of terminology is used in the area of certification and accreditation. In NASA, the term evaluation is used to identify the technical evaluation of the security of an application (synonymous with the FIPS PUB 102 term certification), and certification is used to identify the official management authorization for operation of the application (synonymous with the FIPS PUB 102 term accreditation).

FIPS PUB 102 defines the certification (evaluation) process as consisting of five activities: planning, data collection, basic evaluation, detailed evaluation, and report of findings. The process is summarized in Figure 2-6. The certification (evaluation) process is an iterative process. That is, based on findings from each stage, previous states might have to be reentered and work performed over. For example, basic evaluation might identify a function that is not included within evaluation boundaries but is important for security. This can require revision of the boundaries defined during

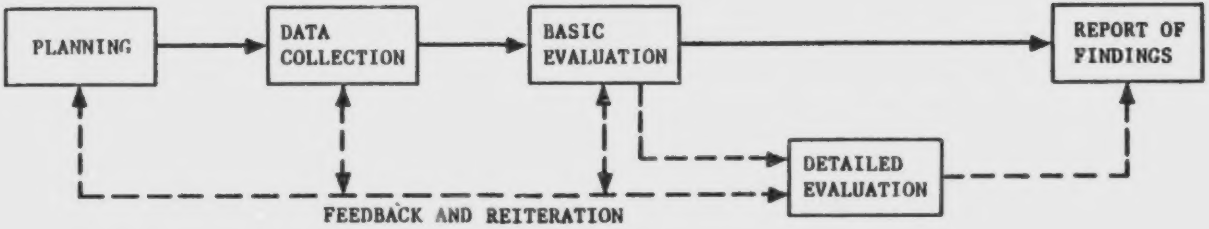

PLANNING
DATA COLLECTION
BASIC EVALUATION
REPORT OF FINDINGS
DETAILED EVALUATION
FEEDBACK AND REITERATION

FIGURE 2-6
CERTIFICATION PROCESS

planning, along with additional data collection. The work is not sequential as suggested in the figure. Typically, most or all stages are ongoing at the same time. The intent of the figure is to show the shift in emphasis as work progresses.

It should be noted that basic evaluation or general evaluation is the minimum necessary for certification (evaluation) to take place. In general, basic evaluation suffices for most aspects of an application under review. However, most certifications (evaluation) also require detailed work in problem areas, and therefore require detailed evaluation as well. (For NASA, security tests are considered to be a detailed evaluation.) Minimum products required for certification and accreditation are a security evaluation report and an accreditation report.

2.3.1.7 Design the Security Specifications

The design stage is the time to make detailed decisions about how the security requirements will be implemented. There are usually a variety of ways to achieve an adequate level of security. In designing security controls, the age-old maxim of "Keep It Simple" is most applicable. A primary source of security problems is excessively complex design that cannot be implemented easily or correctly, and cannot be maintained nor audited. Lonsonsky [15] and Wong [16] suggest that, for any sensitive application, a thorough risk analysis, including safeguard selection, should be performed at the beginning of the design phase to assure that appropriate cost-effective controls are integral to the system's design. The guidance in FIPS PUB 65 [17] can be tailored for a risk analysis of an application system design [2]. FIPS PUB 73 [2] provides some ideas that apply to the overall security design effort.

- Unnecessary Programming. Terminals should be interfaced with the application system so as to minimize the danger that users can get unneeded programming capability. Users who can execute their own programs usually have the potential to bypass any security controls.

- Restricted User Interfaces. User interfaces should be tailored as specifically as possible to fulfill the user's requirements. Unneeded flexibility makes it more difficult to train users and to get them to accept the system. Flexibility also hurts security. The greatest danger occurs when users are given unnecessary access to a general purpose programming language; ...Once the user's needs are understood, interfaces should be designed to meet these needs as simply as possible with no unnecessary capabilities that complicate things both for the users and for the security analyst.

- 'Human Engineering. To preserve security and integrity, user interfaces must be designed so they are easy to understand and use. This can forestall many user errors, and it decreases the chances that users will neglect or bypass controls which they view as cumbersome and annoying.

- Shared Computer Facilities. It is easier to protect the code and data of the application system if it does not share computer facilities with other applications. It is especially useful to exclude all program development activities from the machine that runs the application-including the development and maintenance of the application's programs themselves.

- Isolation of Critical Code. The code and system data that is critical to security should be well identified so it can be more easily audited and protected. When possible, security controls should be isolated in modules that have few interactions with the rest of the application software. This makes it easier to audit these modules and protect them from unauthorized modifications during operations. Sometimes automated controls can be used to protect these modules:

* Checksums of the object code can be used to try to detect unauthorized changes.

* Hardware protection states or protection domains can protect code and data critical to security.

* If security-critical code always resides in a fixed area of memory, then read-only memory can be used.

* Recent experimental languages, called type safe languages, may soon be available. Compilers for these languages can protect modules and their data from unexpected interactions with other modules.

Identification and partial isolation of critical code and data are reasonably easy; however, rigorous isolation is more difficult than it sounds. Without very careful planning, all system code and data will end up being relevant to security because errors or deliberate traps elsewhere can still cause security failures. This other software includes:

* The operating system and other software that supports any of the security functions.

* All parts of the application system needed to guarantee that the security controls are invoked at the appropriate time.

* Compilers and other software used to develop and maintain any security-relevant software.

When code relevant to security is rigorously isolated from the bulk of the software, it is called a security kernel. Security kernels that protect data from unauthorized disclosure are feasible, but they require specially designed operating systems.

- [redacted]

- Use of Available Controls. The operating system and the facility management may already provide a variety of controls such as:

  - User identity verification.
  - Authorization for access to system files.
  - Journaling of operating system activities.
  - Backup and recovery operations.

  While the application system usually needs to supplement these controls, available controls should be used to the fullest extent possible. In too many cases, controls are needlessly reimplemented because controls that are available are not understood or not utilized.

  Since the security controls of an operating system are not absolutely reliable, the application system should use some data integrity checks to try to identify whether critical data has been altered; however, in general, there is no reason to believe that controls implemented in the application will be any more reliable than those already provided by an operating system.

The purpose of the design stage is to translate the security functional requirements into security specifications that can be used by programmers to develop the security-relevant code. Freeman [18] offers the following comments on the purpose of technical design.

...three basic purposes of design can be discerned...:

- discovery of problem structure;
- creation of the outlines (architecture, logical structure) of a solution for the problem;
- review of the results to ascertain if they meet the stated goals.

Once we understand the problem, the next major step is to develop outlines of the solution. This is the creative aspect of design in the strict sense of the word, although developing an accurate understanding of the problem requires just as much creativity in many cases.

The major activity is the establishment of the architecture of the system. That is, we engage in a combination of spelling out, in general terms, how the artifact will look to the user—the functions it will perform--and how it will be built--the major algorithms and data representations it will use.

Some parts of this process of spelling out the overall structure may require that we extend the design to a very detailed level in order to determine the feasibility of performing certain functions. But, in general, we are establishing the major pieces of the system, their relationships, interfaces to other systems and the outside world, and carefully specifying what must be done along with rough indications of how it is to be done.

The third purpose of design is to review repeatedly what has been done so far, to compare it to what is desired, and thus to evaluate progress. Review takes place at all stages of the development cycle, of course, but it is most central to the design phase. Review of code production is intended to determine that what has been implemented is what was specified; review of test results is meant to confirm that a sufficient set of tests has been run; review of specifications seeks to determine if the loosely stated requirements of the customer have been captured in operational terms. Review at the design phase, though, goes beyond just determining if something that has been previously spelled out has

> been done--it is an integral part of the process of discovering the nature of the problem and the proper structure of the solution.

Freeman goes on to describe the software lifecycle as consisting of six stages: analysis, functional specification, architectural design, detailed design, implementation and evolution. Freeman's analysis of the development process, particularly the architectural design and detailed design are appropriate to understanding how to develop specifications for security requirements.

> ...For each stage, we will list the primary inputs (1), outputs (O), and major operations (OP)...
>
> Architectural Design
>
> I: specifications, general context of desired system, knowledge of similar systems
>
> O: structural description of inside of system (definition of modules and interfaces)
>
> OP: discovery of problem structure, identification of major pieces of system, establishment of relationships between parts, abstraction, decomposition
>
> Detailed Design
>
> I: architectural description, programming environment details
>
> O: blueprints for programs
>
> OP: abstraction, elaboration, choice of alternatives

In developing security specifications, there should be a normal evolution from architectural specifications to design specifications. Whether there will be an explicit distinction between architectural-type specifications and detailed specifications will depend on each organization's or Center's standards for software life cycle development. It should be noted that the terms logical design and physical design, used in some methodologies, correspond to SDLC architectural design and detailed design, respectively.

In Section 2.3.1.5, examples of security requirements were provided based on the basic controls described in FIPS PUB 73 [2]. The types of basic controls included: data validation, user identity verification, authorization, journaling, variance detection, and encryption. The following discussion will provide examples of specifications for some of the security requirements previously discussed.

- Requirement - All source data will be keyed twice and automatically compared with the transcribed source data previously keyed.
- Specification - A second person will key the data into a verifier. Only those fields containing transaction code, employee name, SSN, and grade will be verified. Any record containing a discrepancy between the initial keying and the verifying keying will be recorded on a discrepancy file. The discrepancy file will be forwarded to the input control group. Each record containing an error will be visually compared to the original source document and the input control group will resolve any discrepancy. The corrected transaction record will then be submitted input to the system.

  (The operations manual would indicate the record positions of the fields to be verified and the specific instructions for incorporating the correct record/transaction into the batching process.)

- Requirement - All keyed transactions (or transaction of a certain type) will be visually verified prior to transmission to the system.

- Specification - Data entry personnel will visually verify that the transaction code is equal to a 1, 2, or 3; the name field contains no arabic numeric data (note romanic number such as I, II, III, etc., are legal); the SSN field contains no alpha or special characters, no blanks; and that the grade field contains no alpha, special characters or blanks; that grade field contains only one of the following: 01, 03, 04, 05, 07, 09, 10, 11, 13, or 14.

  If any field is not correct, the operator will check the source document for the correct data. If the source document is in error, the document will be returned to the point of origination for correction and resubmission.

- Requirement - All data transmitted between the host computer and remote site will be protected from unauthorized disclosure and modification during transmission.

- Specification - All data will be encrypted using the Data Encryption Standard as specified in FIPS PUB 46.

After the security specifications have been developed and approved, the development of test procedures can begin.

2.3.1.8 Develop Security Test Procedures

At this point in the life cycle, following the definition of the security specifications, Section 3 and 4 of the Security Test Plan can be developed. Using the test plan format at

Appendix A, the next step is to complete the sections dealing with methods and constraints and evaluation.

The methodology should indicate whether static or dynamic testing or both will be used, whether live or test data and an indication of the volume of data that is required to adequately test the security controls. The method for recording test results should be identified as well as any constraints that may limit the scope of the test. Under the evaluation section, the criteria for each type of test should be identified. Data reduction methods should also be described.

FIPS PUB 73 [2] offers some guidance on static and dynamic testing that is useful to this discussion of developing test procedures.

Static Evaluation. These techniques, which involve examination and analysis of the systems documentation and code, represent the most effective way to detect deliberate traps or other unauthorized modifications. However, due to the complexity of most systems and the limitations of automated techniques, and tools, it is not currently practical to analyze systems completely using static evaluation methods. In addition, static evaluation does not examine the system in a "live" or operational mode so that errors in the execution environment are not detected. Specific techniques and tools include:

a. Code Review. Portions of the source code are evaluated to determine if they implement the design specifications and are free from errors. In most cases, it will be impossible to review all of the system's code. Generally, samples of code will be reviewed—especially critical modules or critical portions of the code. The code review can be done by an internal test and evaluation team that is involved in the system development or by an independent (third party) team—either internal or external to the organization. Code review differs from peer review, ...in that the code review is performed by individuals who were not involved with the design and programming of the application.

- **Penetration Studies.** A few individuals can be challenged to find unknown weaknesses in the security controls. Penetration studies to identify programming errors can be expensive and are useful only if someone believes that there are no remaining errors that can affect security.

- **Source Code Analyzers.** These software tools aid the evaluation process by providing details about specific characteristics of the source program.

  Examples include:

  - cross reference listings are an aid to code review and may be useful to identify "suspicious" variables and source statement references.

  - the variables which can influence control flow decisions can all be identified or variables which could be read before any value has been assigned to them can be identified.

> **Dynamic Testing.** These techniques involve executing the application system, or portions of the system, with test data and comparing the actual results with expected or known results. ...Dynamic testing is only practical for selected test cases. Fundamental questions such as, "Which test cases should be chosen?" and "How many test cases are enough?" must be answered. The answers to these questions depend upon the system being tested and upon the experience of the test team.
>
> The following tools can be used to aid the dynamic testing process:
>
> * **Program Analyzers.** A program analyzer is a computer program that collects data about another program's operation while that program is executing. Program analyzers can be particularly useful for evaluating how thoroughly the test data has exercised the program being tested. In addition, they can be used to identify extraneous code that might be an unauthorized insertion.
>
> * **Flaw Hypothesis Method.** Security flaws can be hypothesized based on analogous flaws found in other systems and then tested for existence in this system. This is an effective approach for selecting test cases that are likely to find flaws.

The following excerpts for FIPS PUB 102 [8] on basic and detailed evaluation provide guidance that should be useful in developing test procedures.

...The general distinction between basic and detailed evaluation is that basic evaluation is primarily concerned with the overall functional security posture, not with the specific quality of individual controls. ...Basic evaluation is also concerned with verifying that security functions actually exist and that the implementation method is of sufficient quality to be relied upon. Detailed evaluation, on the other hand, is concerned with whether security functions work properly, satisfy performance criteria, and acceptably resist penetration.

There are four tasks in basic evaluation:

1. security requirements evaluation (are application security requirements acceptable?)

2. security function evaluation (do application security functions satisfy requirements?)

3. control existence determination (do the security functions exist?)

4. methodology review (does the implementation method provide assurance that security functions are acceptably implemented?)

Security Requirements Evaluation. In both formulating and evaluating security requirements for an application, two classes of needs are considered: policy needs and situational needs. Policy needs derive from the principles and required practices that the application is obliged to pursue, such as Federal laws, regulations, standards and agency policies. Situational needs are those deriving from the application's characteristics and environment. To determine situational needs, four primary areas are considered: assets, threats, exposure and controls.

1. Asset. What should be protected?

2. Threats. What are assets being protected against?

3. Exposures. What might happen to assets if a threat is realized?

4. Controls. How effective are security safeguards in reducing exposures?

Security Function Evaluation. Given well-defined security requirements, function evaluation becomes the most important task in basic evaluation. It determines whether security functions (control techniques) such a authentication, authorization, monitoring, security management, and security labeling satisfy security requirements. The primary method is simply to use the stated requirements as a checklist to follow in assessing whether they are satisfied.

In some cases, requirements specify only the need for generic functions such as authentication. In other cases the requirements call for use of specific mechanism, such as particular password technique. In both situations, function evaluation identifies the defined security function and examines it for acceptability.

An important concern for function evaluation is the appropriate level of detail. The recommendation is that basic evaluations be complete (all applicable control features) down through the functional level, where "functional level" is the logical level represented by functions as defined in (or appropriate for definition) in the Functional Requirements Document. This notion applies to both controls within the computer and physical/administrative controls external to it (although the latter might not actually be defined in a Functional Requirements Document).

Control Existence Determination. The fact that functions are described in a document or discussed in an interview does not prove that they have been implemented. Basic evaluations require assurance that security function controls exist. The existence of most physical and administrative controls can be determined via visual inspection. For controls internal to the computer, testing is needed. ...The intent is to simply verify that the functions exist.

Test for control existence determination are straight-forward. In many cases, a short operational demonstration suffices. For example, the existence of a password function can be determined by attempting to use the application and verifying that a valid password is required.

Methodology Review. Even though this is a high-level overview-type evaluation, it is still desirable to gain some assurance that controls are acceptably implemented. The best way to do this without becoming immersed in testing or detailed analysis is to examine the methodology used to develop the application.

The areas of concern in reviewing a development methodology for certification are summarized below.

1. Documentation. Is there current, complete and acceptable-quality documentation?

2. Objectives. Was security explicitly stated and treated as an objective, with an appropriate amount of emphasis for the situation? Were security requirements defined?

3. Project Control. Was development well-controlled? Were independent review and testing performed and did they consider security? Was an effective change control program used?

4. Tools and Techniques. Were structure design techniques used (e.g., modularization, formal specifications)? Were established programming practices and standards used (e.g., high order languages, structured walk-throughs)?

5. Resources. How experienced in security were the people who developed the application? What were the sensitivity levels or clearances associated with their positions?

Detailed evaluations involve analysis of the quality of security safeguards. Primary tasks are examinations of the application from three points of view:

1. Functional Operation (Do controls function properly?)

2. Performance (Do controls satisfy performance criteria?)

3. Penetration Resistence (How readily can controls be broken or circumvented?)

Detailed evaluation consists of a collection of approaches. Selection of which to use depends primarily on the threats and exposures of concern, rather than on the general characteristics or overall sensitivity of the application. To illustrate, if the primary concern is to protect secrets from an external penetrator, penetration resistance is stressed.

Functional Operation. Functional operation is the point of view most often emphasized in detailed evaluation since it assesses protection against human errors and casual attempts to misuse the application. Evaluations of function operation assess whether controls acceptably perform their required functions. Although testing is the primary technique in evaluating functional operation, other validation and verification techniques must also be used, particularly to provide adequate analysis and review in early phases of the application life cycle. Testing for functional operation examine areas such as the following:

1. Control operation (e.g., do controls work?)
2. Parameter checking (e.g., are invalid or improbable parameters detected and properly handled?)
3. Common error conditions (e.g., are invalid or out-of-sequence commands detected and properly handled?)
4. Control monitoring (e.g., are security events such as errors and file accesses properly recorded; are performance measurement of characteristics such as resource utilization and response time properly recorded?)
5. Control management (e.g., do security procedures for changing the security table work?)

To illustrate this testing, consider several of the tests needed to examine control operation of a password function:

1. Test whether access without a password is disallowed.
2. Test whether valid passwords are accepted and invalid passwords are rejected.
3. ...Test the interface between the password function and the access authorization function by testing

whether access is properly allowed or disallowed. For example, verify that valid passwords allow proper access and do not allow improper access, and that invalid passwords result in proper access restriction.

4. Test whether the system responds correctly to multiple invalid passwords.

5. Test whether system-initiated reauthentication functions correctly.

Functional operation includes the application's resistance to external errors. Therefore the test areas of primary interest include those interfaces across which errors might propagate:

1. man-man (e.g., operator messages)

2. man-system (e.g., commands, procedures)

3. system-system (e.g., intersystem dialogue)

4. process-system (e.g., calls)

5. process-process (e.g., interprocess calls)

Besides testing, there are other security evaluation tools and techniques that can be of use in examining functional operations. For example, software tools for program analysis, can be helpful in documentation analysis. Matrices can suggest ideas for test cases and scenarios. Checklists have utility in providing quick training as well as suggesting ideas for tests.

Formal verification is a technique that may be used during a detailed evaluation. Formal verification offers the hope of being able to mathematically "prove" that a functional design abides by a few simple security rules, and that lower levels of abstraction are consistent with the proven higher-level design performance. A number of qualitative factors are listed under the general heading of performance, which is the second area of concern in detailed evaluation. These are availability, survivability, accuracy, response time, and throughput. They can be applied to either individual controls or entire applications.

1. Availability. What proportion of time is the application available to perform critical or full services: Availability incorporates many aspects of reliability, redundancy, and maintainability. It is often more important than accuracy. It is especially relevant to applications with denial of service exposures as primary concerns (e.g., air traffic control, automatic funds disbursement, production control). Security controls usually require higher availability than other portions of an application.

2. Survivability. How well does the application or control withstand major failures or natural disasters? "Withstand" includes the support of emergency operations afterwards, and recovery actions to return to normal operation.

3. Accuracy. How accurate is the application or control? Accuracy encompasses the number, frequency and significance of errors. Controls for which accuracy measures are especially applicable are identity verification techniques and communication line handling techniques.

4. Response Time. Are response times acceptable? Slow control response time can entice users to bypass the controls. Examples of controls for which response time is critical are passwords (especially in distributed networks) and identity verification techniques.

5. Throughput. Does the application or control support required usage capacities? Capacity includes the peak and average loading of such things as users and service requests.

Penetration Resistance. The task here is to assess resistance against the breaking or circumventing of controls, where resistance is the extent to which the application and controls must block or delay attacks.

Assessment of penetration resistance can be the most technically complex of the detailed evaluation categories. It is best done to establish confidence in security safeguards. It can also be done to find and fix flaws. In both cases it:

1. provides an assessment of an application's penetration resistance;
2. helps to determine the difficulties involved in actually exploiting flaws; and
3. provides a clear demonstration of flaw exploitability (since it might not be clear from analysis whether, say, an asynchronous timing flaw can be exploited).

The objective of penetration-resistance evaluation is to identify externally exploitable flaws in internal security functions and the interfaces to them. Following are illustrative areas for this detailed examination:

1. complex interfaces
2. change control process
3. limits and prohibitions
4. error handling
5. side effects
6. dependencies
7. design modifications/extensions
8. control on security descriptors
9. execution chain of security services
10. access to residual information

Additional information on these areas can be found in the IBM Systems Journal paper entitled: "Penetrating an Operating System: A Study of VM/370 Integrity" [19].

The finalization of the test procedures will be a simultaneous activity by the test team while system developers are writing the security code and documenting the security safeguards.

2.3.1.9 Write Security Relevant Code

The precise way in which security safeguards will actually be implemented in the software code will depend to a great extent on the programming language used, installation standards, and personal programming style. Within the foregoing limitations there are some practices that should be followed to ensure that security relevant code is understandable, auditable, maintainable and testable.

The System Auditability and Control (SAC) Study [3] in discussing application system development controls provides some guidance that is useful to the development of security relevant code.

> The adequacy and effectiveness of controls included in computer application systems are affected by the methods and procedures used during the system development process.

One of the areas discussed in some depth is the use of structured programming. The following is a summary of the SAC discussion of structured programming.

> The objective in using structured programming techniques is to develop more usable and effective programs. "Usable" implies that the program can be read and understood by technical persons who did not write it, including users and EDP auditors. "Effective" implies that the program is designed to fit into an overall application system scheme so as to reduce redundancy and ensure processing efficiency.*
>
> Structured programming is a technique for system builders that renders systems easier to build, maintain and alter. It is a discipline that is used primarily in the detail design and

> programming states of the development process. As such, it uses a stepwise top-down approach, in which program modules are organized by functional specifications into a balanced hierarchical structure with minimum side effects on each module.
>
> Structured programming involves a team approach to detailed design, with team members being used to "walk-through" the design and coding of components. The effect is that the design and code can be viewed by other than the originator to detect faulty logic, hard-to-follow code and the extent to which the design meets prespecified objectives.

Stanford Research Institute, authors of the SAC study, found the following major techniques were used by many organizations:

- Program Structure. A semistrict program structure allowing GO TO statements in a downward direction within a section domain.
- Statement Formatting. Fixed column indentation for both processor division and data division sections; in addition, a maximum of one verb per line and specific columns for operators.
- Peer Reviews. Structured walk-throughs whereby at least two peers completely trace or walk through the code generated by another programmer.
- Team Organization. The establishment of an integrated team consisting of one project leader/analyst, two programmer analysts, one to three programmers and one programmer librarian.

---

* Security controls should meet the usable objective. To be effective, in the above context, security controls should not unnecessarily reduce processing efficiency and be implemented in a way that they are part of the overall application scheme, and do not appear or operate as an interruption to the normal flow of the program or application processes.

- Top-Down Design. This technique consists of designing program logic by specifying higher level functions first and determining the subfunctions required to implement these higher level functions.

- Segmentation. During detail design and programmng, it is advantageous to keep programs and modules in the form of routines called segments, with each segment having but one entry and exit.*

- Structured Coding. This approach or discipline is used to depict the process of coding whereby there are conventions used for syntax, program format, restricted and controlled branching, and disciplines on logic.

- Walk-throughs. The walk-through consists of a planned review of all system specifications and coding by peers of the developers. Walk-throughs have been found to be instrumental in uncovering a majority of errors during the pre-installation and test phases of a system. It is usually worthwhile to allow others to review specifications and code before a joint meeting. The review meeting can then become mainly a question answering and resolution session.

- Programmer Librarian. The programmer librarian . actually serves the purpose of documenting all source codes. This function is responsible for getting codes keyed, updating the source library, and other general program documentation.

FIPS PUB 73 [2] recommends the following practices to enhance the security of application systems:

- Program Libraries. The program library catalogs and controls access to all versions of program modules as they are being developed. These control functions can be carried out by either manual or automated means. The program library can provide the following types of security controls during the programming stage:

---

* The segmentation technique is particularly applicable to the coding of security relevant code.

* permit only authorized persons access to program modules,

* record all accesses (especially modifications) to program modules,

* associate control data, such as record and byte counts, with program modules to facilitate detection of changes, and

* enable comparison of current versions of modules with previous versions to identify code that has been changed.

More rigorous controls are needed as the program modules near completion, especially once review and testing has begun.

o Redundant Computation. Critical computations can be checked by redundant processing to verify correctness of the result. Examples of local redundancy checks include:

  * recalculation of a critical result by an alternate method;

  * checking a calculated result for reasonableness and consistency with other data items; and

  * examining extra attributes in retrieved data to ensure that the data item found was the one that was searched for.

- Program Development Tools. The choice of the programming language and of other programming tools can enhance the reliability and correctness of the final products. Proper selection and utilization of such tools will help prevent programming errors from entering the source code. Some specific program development tools include:

- High Level Programming Languages. Programming languages are especially useful if they support structured control flow, extensive data definition facilities, strong type checking, restricted scopes for program variables, and well-defined module calling conventions. Compilers for such languages can do extensive checking to identify program errors.
- Preprocessors. Many of the advantages of high-level programming languages can be accomplished through a preprocessor. A preprocessor can be used to:
  - eliminate some of the more restrictive conditions in an existing language (e.g., allow structured flow in FORTRAN programs), and
  - provide automated quality control by checking that program modules meet the coding standards of this project.
  - Other Tools. Program development tools such as those that reformat source code, produce cross-reference listings and aid debugging are useful to help programmers manage the complexities.

2.3.1.10 Document Security Safeguards

Program documentation is needed during any software development; it is especially necessary for security-relevant code. FIPS PUB 73 [2] defines security-relevant code as:

- code that implements security controls;
- code that performs critical processing (e.g., check disbursement, real-time controls); and
- code that has access to critical or sensitive data during its execution.

Brill [13] discusses the documentation of security controls as follows:

> You plan your controls; verify them through reviews with users and management; verify the operability of the controls through tests where you can; and make whatever modifications are necessary to get the best overall workable set of controls. Then make sure—again by review and test—that your documentation properly reflects the controls you want and the way that you want them to work.

The SAC [3] study provides the following thoughts on documentation that are applicable to the documentation of security controls.

> Documentation is the process of describing on paper what functions an application system performs, how it performs them, and how the functions are to be used. The objectives of good documentation are to provide application system designers, implementers, testers, users and EDP auditors with a clear means of understanding all aspects of the application system.
>
> Documentation...is important because it helps ensure correct and efficient processing within both data processing and user areas; it increases the ease and accuracy of computer program maintenance, and it provides auditors with an independent basis for evaluating application control.
>
> One approach to documentation is used by a large food manufacturer. This organization makes use of a documentation test to be administered throughout the phases of system development. The primary purpose is to ensure that appropriate documentation exists for the major phases of system development;...
>
> In addition to verifying the existence of sufficient documentation, this technique also assesses various aspects of the documentation, including the following:
>
> - Does the documentation give evidence that processing controls will be adequate?

- Have sufficient controls been built into the system to allow effective operation and maintenance?
- Can the documentation be used as a basis to prove that controls over operation and maintenance are adequate?

It is recommended that the documentation of security-relevant code be contained in an independent document. The sensitivity of security-relevant code is such that it should be well-protected and access to the documentation should be restricted.

2.3.1.11 Conduct Security Test and Evaluation

Ideally, the conduct of the actual security test and evaluation will be performed by an organization independent of the system developers and users; that is, an independent validation and verification (IV&V) team, quality assurance personnel or an audit group. If such groups do not formally exist within the organization a team of security evaluators may be formed for the purpose of conducting the security test and evaluation. Team skills that will be required include application analysts, testers, programmers, penetration specialists, VV&T specialists and security.

The actual test will consists of some combination of the following: document reviews, interviews, dynamic tests and penetration resistance tests. Each of these have been previously discussed.

Regardless of what combination of tests is used, it is important that the test team keep a list of documents reviewed, document all interviews, and document and maintain the results of any tests run against the application.

2.3.1.12 Write Security Test and Evaluation Report

The purpose of the Security Test and Evaluation Report is to document the test and evaluation results and findings; present the demonstrated capabilities and deficiencies of the security controls; and provide a basis for preparing the proposed security certification report.

The Security Test and Evaluation Report should be prepared by, or under the direction of, the security team leader. It will then be transmitted to the application computer security official (CSO). FIPS PUB 103 [8], provides a sample outline that can be used to develop the Security Test and Evaluation Report (see Figure 2-7). As noted in Figure 2-8, the sample outline contains five sections. Section 1, Introduction and Summary, briefly describes the application and summarizes the evaluation findings and recommendations. Section 2, Background, provides contextual information for the Application CSO including the security standards and policies that were applied. Also, it should include a list of the general functional characteristics of the application that generally influence its certifiability (e.g., the absence or presence of user programming). The scope of the evaluation and the assumptions and constraints on the test should be identified.

Section 3, Major Findings, should summarize the controls in place and their role in protecting the data and/or the application. It should emphasize those controls that were found to be effective. Any vulnerabilities found during the test should also be documented. The report should identify those vulnerabilities which should be accepted and those which should be corrected.

1. INTRODUCTION AND SUMMARY

2. BACKGROUND

3. MAJOR FINDINGS

   3.1 General Control Posture
   3.2 Vulnerabilities

4. RECOMMENDED CORRECTIVE ACTIONS

5. EVALUATION PROCESS

Attachment A: Proposed Certification Statement

FIGURE 2-7
SAMPLE OUTLINE FOR A SECURITY EVALUATION REPORT

Section 4, Recommended Corrective Actions, identifies what additional controls should be considered and the anticipated costs of such recommendations. Section 5, Evaluation Process, should summarize the security test and evaluation process to include the types of tests that were conducted.

2.3.1.13 Prepare the Proposed Certification Statement

The proposed Certification Statement should summarize the recommendations, the acceptability of applications' security safeguards, restrictions (e.g., applications must be run as a stand-alone system), and/or corrective actions that must be accomplished prior to allowing the application to commence running.

ATTACHMENT 1

EXECUTIVE OFFICE OF THE PRESIDENT
OFFICE OF MANAGEMENT AND BUDGET
WASHINGTON, D.C. 20503

July 27, 1978

CIRCULAR NO. A-71
Transmittal Memorandum No. 1

TO THE HEADS OF EXECUTIVE DEPARTMENTS AND ESTABLISHMENTS

SUBJECT: Security of Federal automated information systems

1. Purpose. This Transmittal Memorandum to OMB Circular No. A-71 dated March 6, 1965 promulgates policy and responsibilities for the development and implementation of computer security programs by executive branch departments and agencies. More specifically, It:

a. Defines the division of responsibility for computer security between line operating agencies and the Department of Commerce, the General Services Administration, and the Civil Service Commission.

b. Establishes requirements for the development of management controls to safeguard personal, proprietary and other sensitive data in automated systems.

c. Establishes a requirement for agencies to implement a computer security program and defines a minimum set of controls to be incorporated into each agency computer security program.

d. Requires the Department of Commerce to develop and issue computer security standards and guidelines.

e. Requires the General Services Administration to issue policies and regulations for the physical security of computer rooms consistent with standards and guidelines issued by the Department of Commerce; assure that agency procurement requests for automated data processing equipment, software, and related services include security requirements; and assure that all procurements made by GSA meet the security requirements established by the user agency.

f. Requires the Civil Service Commission to establish personnel security policies for Federal personnel associated with the design, operation or maintenance of Federal computer systems, or having access to data in Federal computer systems.

2. Background. Increasing use of computer and communications technology to improve the effectiveness of governmental programs has introduced a variety of new management problems. Many public concerns have been raised in regard to the risks associated with automated processing of personal, proprietary or other sensitive data. Problems have been encountered in the misuse of computer and communications technology to perpetrate crime. In other cases, inadequate administrative practices along with poorly designed computer systems have resulted in improper payments, unnecessary purchases or other improper actions. The policies and responsibilities for computer security established by this Transmittal Memorandum supplement policies currently contained in OMB Circular No. A-71.

3. Definitions. The following definitions apply for the purposes of this memorandum:

a. "Automated decisionmaking systems" are computer applications which issue checks, requisition supplies or perform similar functions based on programmed criteria, with little human intervention.

b. "Contingency plans" are plans for emergency response, back-up operations and post-disaster recovery.

c. "Security specifications" are a detailed description of the safeguards required to protect a sensitive computer application.

d. "Sensitive application" is a computer application which requires a degree of protection because it processes sensitive data or because of the risk and magnitude of loss or harm that could result from improper operation or deliberate manipulation of the application (e.g., automated decisionmaking systems).

e. "Sensitive data" is data which requires a degree of protection due to the risk and magnitude of loss or harm which could result from inadvertent or deliberate disclosure, alteration, or destruction of the data (e.g., personal data, proprietary data).

Resp of Management ①

4. Responsibilities of the heads of executive agencies. The head of each executive branch department and agency is

responsible for assuring an adequate level of security for all agency data whether processed in-house or commercially. This includes responsibility for the establishment of physical, administrative and technical safeguards required to adequately protect personal, proprietary or other sensitive data not subject to national security regulations, as well as national security data. It also includes responsibility for assuring that automated processes operate effectively and accurately. In fulfilling this responsibility each agency head shall establish policies and procedures and assign responsibility for the development, implementation and operation of an agency computer security program. The agency's computer security program shall be consistent with all Federal policies, procedures and standards issued by the Office of Management and Budget, the General Services Administration, the Department of Commerce, and the Civil Service Commission. In consideration of problems which have been identified in relation to existing practices, each agency's computer security program shall at a minimum:

a. Assign responsibility for the security of each computer installation operated by the agency, including installations operated directly by or on behalf of the agency (e.g., government-owned contractor operated facilities), to a management official knowledgeable in data processing and security matters.

b. Establish personnel security policies for screening all individuals participating in the design, operation or maintenance of Federal computer systems or having access to data in Federal computer systems. The level of screening required by these policies should vary from minimal checks to full background investigations commensurate with the sensitivity of the data to be handled and the risk and magnitude of loss or harm that could be caused by the individual. These policies should be established for government and contractor personnel. Personnel security policies for Federal employees shall be consistent with policies issued by the Civil Service Commission.

c. Establish a management control process to assure that appropriate administrative, physical and technical safeguards are incorporated into all new computer applications and significant modifications to existing computer applications. This control process should evaluate the sensitivity of each application. For sensitive applications, particularly those which will process sensitive data or which will have a high potential for loss, such as automated decisionmaking systems, specific controls should, at a minimum, include policies and responsibilities for:

(1) Defining and approving security specifications prior to programming the applications or changes. The views and recommendations of the computer user organization, the computer installation and the individual responsible for the security of the computer installation shall be sought and considered prior to the approval of the security specifications for the application.

Testing

(2) [redacted] The objective of the design reviews should be to ascertain that the proposed design meets the approved security specifications. The objective of the system tests should be to verify that the planned administrative, physical and technical security requirements are operationally adequate prior to the use of the system. The results of the design review and system test shall be fully documented and maintained as a part of the official records of the agency. Upon completion of the system test, an official of the agency shall certify that the system meets the documented and approved system security specifications, meets all applicable Federal policies, regulations and standards, and that the results of the test demonstrate that the security provisions are adequate for the application.

Audits

d. [redacted] sensitive application including those which process personal, proprietary or other sensitive data, or which have a high potential for financial loss, such as automated decisionmaking applications. Audits or evaluations are to be conducted by an organization independent of the user organization and computer facility manager. Recertifications should be fully documented and maintained as a part of the official documents of the agency. Audits or evaluations and recertifications shall be performed at time intervals determined by the agency, commensurate with the sensitivity of information processed and the risk and magnitude of loss or harm that could result from the application operating improperly, but shall be conducted at least every three years.

e. Establish policies and responsibilities to assure that appropriate security requirements are included in specifications for the acquisition or operation of computer facilities, equipment, software packages, or related services, whether procured by the agency or by the General Services Administration. These requirements shall be reviewed and approved by the management official assigned responsibility for security of the computer installation to be used. This individual must certify that the security requirements specified are reasonably sufficient for the intended application and that they comply with current Federal computer security policies, procedures, standards and guidelines.

Risk Analysis

f. [redacted] The objective of this risk analysis should be to provide a measure of the relative vulnerabilities at the installation so that security resources can effectively be distributed to minimize the potential loss. A risk analysis shall be performed:

(1) Prior to the approval of design specifications for new computer installations.

(2) Whenever there is a significant change to the physical facility, hardware or software at a computer installation. Agency criteria for defining significant changes shall be commensurate with the sensitivity of the information processed by the installation.

(3) At periodic intervals of time established by the agency, commensurate with the sensitivity of the information processed by the installation, but not to exceed five years, if no risk analysis has been performed during that time.

Contingency Planning

g. [redacted] The objective of these plans should be to provide reasonable continuity of data processing support should events occur which prevent normal operations. These plans should be reviewed and tested at periodic intervals of time commensurate with the risk and magnitude of loss or harm which could result from disruption of data processing support.

5. Responsibility of the Department of Commerce. The Secretary of Commerce shall develop and issue standards and

guidelines for assuring security of automated information. Each standard shall, at a minimum, identify:

a. Whether the standard is mandatory or voluntary.

b. Specific implementation actions which agencies are required to take.

c. The time at which implementation is required.

d. A process for monitoring implementation of each standard and evaluating its use.

e. The procedure for agencies to obtain a waiver to the standard and the conditions or criteria under which it may be granted.

6. Responsibility of the General Services Administration. The Administrator of General Services shall:

a. Issue policies and regulations for the physical security of computer rooms in Federal buildings consistent with standards and guidelines issued by the Department of Commerce.

b. Assure that agency procurement requests for computers, software packages, and related services include security requirements which have been certified by a responsible agency official. Delegations of procurement authority to agencies by the General Services Administration under mandatory programs, dollar threshold delegations, certification programs or other so-called blanket delegations shall include requirements for agency specifications and agency certification of security requirements. Other delegations of procurement authority shall require specific agency certification of security requirements as a part of the agency request for delegation of procurement authority.

c. Assure that specifications for computer hardware, software, related services or the construction of computer facilities are consistent with standards and guidelines established by the Secretary of Commerce.

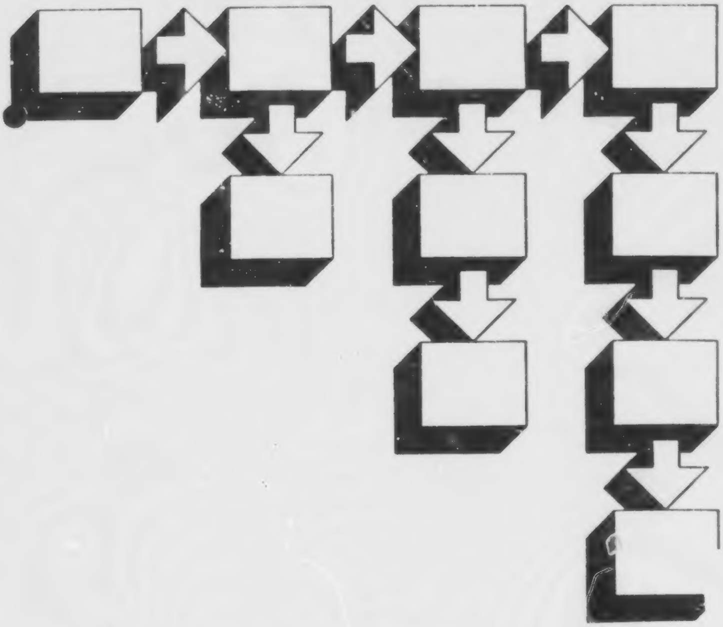
d. Assure that computer equipment, software, computer room construction, guard or custodial services, telecommunications services, and any other related services procured by the General Services Administration meet the security requirements established by the user agency and are

U.S. Department
of Commerce
National Bureau
of Standards

Computer Science and Technology

NBS Special Publication 500-109

Overview of Computer Security Certification and Accreditation



Computer Science and Technology

NBS Special Publication 500-109

Overview of Computer Security Certification and Accreditation

Zella G. Ruthberg

Center for Programming Science and Technology
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, DC 20234

William Neugent

System Development Corporation
7929 Westpark Drive
McLean, VA 22102

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Bureau of Standards
Ernest Ambler, Director

Issued April 1984

FOREWORD

This overview is intended to provide ADP policy managers, information resource managers, ADP technical managers, and ADP staff with a comprehensive summary of and guide to FIPS PUB 102, "Guideline to Computer Security Certification and Accreditation," September 27, 1983. FIPS PUB 102 presents in detail an approach to developing a program and performing a technical process for certifying and accrediting sensitive applications. By summarizing FIPS PUB 102 and referencing its relevant sections, this overview will not only enable its different type audiences to obtain a complete picture of the certification/accreditation activity, but also direct these audiences to parts of FIPS PUB 102 relevant to them. This overview and FIPS PUB 102 should be of particular interest to all those responsible for responding to Office of Management and Budget Circular A-71, Transmittal Memorandum Number 1, July 27, 1978.

SECTION 1

INTRODUCTION

Some computer security risks threaten the very existence of an organization. Critical decisions regarding the adequacy of security safeguards in sensitive applications must be made by authorized managers and must be based on reliable technical information. [redacted] Together they provide management a quality control technique for computer security. [redacted] fraud, illegal practices, [redacted] embarrassing 'leaks'" and legal action. [redacted] This Guideline explains the need for and describes the major features of the certification and accreditation processes.

1.1 PROBLEMS ADDRESSED

Although the introduction of computers into the daily operation of organizations has sometimes obscured the issue, managers are still responsible for protecting the organization's vital resources, including its information systems. Computers affect this security responsibility in two important ways. First, computers can radically change information system vulnerabilities from those found in manual systems and, in many cases, can increase the risks to data and resources. Second, computers increase the complexity of systems, thus making such systems more difficult to understand and protect. Safeguards can be much more complex than those used in manual systems, and the evaluation of these also becomes more complex. Some examples of the changes introduced by computers and the security problems raised by these changes are:

1. Data assets can be more centralized, thus permitting larger scale frauds and magnifying errors.
2. Decisions formerly made by people can be made automatically, making such decisions more susceptible to tampering.

3. Access may be achieved electronically through a remote terminal, rather than directly at the computer site, thus permitting unauthorized and unobserved access more readily.

4. Duties previously separated for security reasons may be integrated in the computer, thus allowing frauds and errors to occur undetected.

5. The computer can become such a critical asset that its failure may cause major organizational disruption.

6. Organizations' management structures are shifting to accommodate computer technology. This often results in fuzzy allocation of management responsibility, creating holes and overlaps in management control.

To counter these problems, managers need techniques to assess and cost-effectively improve their computer security posture. Certification and accreditation, when applied to computer security, are techniques for achieving these ends.

1.2 CONTEXT FOR COMPUTER SECURITY CERTIFICATION AND ACCREDITATION

Computer security certification and accreditation are only one aspect of a general certification and accreditation activity that should be performed to ensure that a computer application satisfies its defined functional, performance, security, and quality requirements. This general process often utilizes the same methods, techniques, and technical tools used for performing technical evaluations for other purposes. The guidance here focuses on those aspects of this general process that are relevant to the computer security of an ADP application. For the remainder of this overview 'computer security certification,' 'security certification,' and 'certification' will be used synonymously as will the terms 'computer security accreditation,''security accreditation,' and "accreditation."

1.3 ENTITIES REQUIRING CERTIFICATION AND ACCREDITATION

Certification and accreditation are only performed on those applications sensitive enough to warrant such attention, where an application's sensitivity derives from the potential loss or harm associated with a security failure during operation (Sec. 1.2.7)[1]. To be cost-effective, this process also requires that an organization form a prioritized list of sensitive applications, based on factors such as mission importance, asset value, and anticipated threats. Some organizations have sensitivity categorization schemes which they use for this prioritizing activity (App. C).

---

[1] Section numbers in parentheses indicate where this material may be found in FIPS PUB 102, "Guideline for Computer Security Certification and Accreditation."

In this discussion 'computer system' and 'computer application' are closely related terms. A computer system is an assembly of elements including at least hardware and usually also software, data, procedures, and people, so related as to behave as an interacting or interdependent unity (Sec. 1.2.4); a computer application is the use(s) for which a computer system is intentionally employed (Sec. 1.2.5).

SECTION 2

WHAT ARE CERTIFICATION AND ACCREDITATION?

[redacted] (Sec. 1.2.2 and 1.2.3)

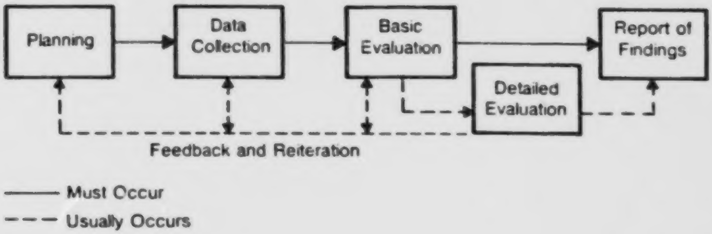

Planning
Data Collection
Basic Evaluation
Report of Findings
Detailed Evaluation
Feedback and Reiteration
Must Occur
Usually Occurs

Figure 1. The Certification Process

2.1 CERTIFICATION PROCESS

Figure 1 summarizes the certification process. It is iterative in that, based on findings from each stage, previous stages might have to be reentered and work performed again. Typically most or all of the stages are ongoing at the same time. The intent of the figure is to show the shift in emphasis as work progresses.

There are two levels of activity associated with security evaluation for certification: basic evaluation and detailed evaluation. Basic evaluation means 'high-level' or 'overview-type' evaluation and is essential in all certifications. Usually basic evaluation suffices for most aspects of an application under review. However, most certifications also require detailed work in problem areas and therefore require some detailed evaluation as well.

Time and resources required to perform a certification vary widely from case to case. If potential loss or harm is low, certification cost must also be kept low. Risk analysis can help in deciding how much certification review is cost justified. Resources for certification may vary from several person-days to many person-months. Minimum products required from certification and accreditation are a security evaluation report and an accreditation statement.

The certification process described here is in a functional form. It tells what must be done and presents a general functional view of how to accomplish it. Detailed specifics of security evaluation for certification will differ widely from case to case and must be adapted to meet specific application needs. Aids such as detailed evaluation methods and checklists are helpful in the adaptation process but no single detailed method exists that can be used universally.

Since the certification process described is at a functional level, it can be applied to both applications under development and those already operational. For example, both could include review of similar documentation such as Functional Requirements Documents and test procedure reports. Detailed evaluation methods differ for the two situations, however, due to differences in the types of data available, the time frame in which data are available, and the organization of the work.

2.1.1 Planning (Sec. 2.1)

Since the planning process must anticipate problem areas and define needs for specialized skills, it requires that the planners perform a very high-level quick review of the entire application in order to gain an understanding of the issues involved. Additional planning tasks are those of (1) placing boundaries on the effort, (2) partitioning the work, (3) identifying areas of emphasis, and (4) drawing up a certification plan.

For certification, application boundaries must be drawn to include all relevant facets of the application's environment including the administrative, physical, and technical areas. Without this comprehensive review, certification gives an incomplete and perhaps misleading picture of application security. For example, excellent technical controls may be rendered worthless if administrative security duties are not properly defined.

Within these overall boundaries, certification work is usually partitioned based on the specialized skills involved. A sample partitioning of security evaluation responsibility areas follows: administrative security, computer operation, contingency planning, change control, data entry and output, operating system, communication security, personnel security, physical security, environmental controls, development method, application software controls, data base management system, and hardware.

The greatest emphasis should be placed on areas of greatest potential loss or harm. These may have been identified in an earlier risk analysis or in reports of past problems or violations. It may also be that the accrediting official(s), based on management judgment, desire emphasis in a particular area.

The information collected in the planning phase forms the basis for the application's certification plan (Sec. 2.1.4). Suggested sections for the plan include security requirements, the evaluation approach, the evaluation team make-up, a schedule, required support, and evaluation products.

2.1.2 Data Collection (Sec. 2.2)

The ideal source of information is application documentation. Critical documents are (1) application system security requirements; (2) a risk analysis showing threats and assets; (3) an application flow diagram showing inputs, processing steps, and outputs, along with complete transaction flows for important transaction types; and (4) a listing of application controls. Unfortunately, few applications have a complete set of this documentation. Where such documents do not exist, the most efficient technique for gathering this information is for application personnel to prepare this documentation and to also provide supplementary tutorial briefings to the certification team. Document reviews of the more commonly found documents (e.g., user manuals) and interviews are also needed to expand upon and corroborate the information in the documents listed above.

2.1.3 Basic Evaluation (Sec. 2.3)

Basic evaluation is primarily concerned with the overall security function posture. For example, it might be concerned with whether authorization subjects must include terminals as well as individuals and processes.

There are four tasks in a basic evaluation:

1. Security Requirements Evaluation (are security requirements acceptable?)

2. Security Function Evaluation (does the design or description of security functions satisfy the security requirements?)

3. Control Implementation Determination (are the security functions implemented?)

4. Methodology Review (does the implementation method provide assurance that security functions are acceptably implemented?)

1. Requirements Evaluation: Requirements evaluation is important because certification is only meaningful if the application has well-defined security requirements. Unfortunately this is often not the case. This task then critically examines any documentation of these requirements and compares it with Federal, state, organizational and user requirements. Where no such documentation exists, the security requirements implied in the application must be formulated.

In both formulating and evaluating security requirements, consideration is given to Federal and state laws and regulations, organizational standards and policies, and the specific application needs. The four primary areas considered in defining application needs are assets, exposures, threats, and controls. Corresponding questions to be answered are: What should be protected? What might happen to assets if a threat is realized? What are assets being protected against? and How effective are security safeguards in reducing exposures?.

If a risk analysis has been performed for the application or its environment, many situational security needs might already be well defined. Other useful tools are computer security checklists and questionnaires, many of which are now available [2].

2. Function Evaluation: Function evaluation determines whether security functions such as authentication, authorization, monitoring, security management, and security labeling satisfy security requirements. With well-defined security requirements, this function evaluation becomes the most important task in basic evaluation. The primary method is to use the stated requirements as a checklist. For example, where called for in requirements: Is individual accountability provided? Are subjects and objects identified and given security labels? Is an execute-only mode of access provided? Are all file accesses recorded? Are functions partitioned so as to provide separation of duties? Does a contingency plan exist and has it been tested?

---

[2] For further information see Ruthberg, Zella G. (Editor), William Neugent, John Gilligan, and Lance Hoffman, "Technology Assessment: Methods for Measuring the Level of Computer Security," NBS Special Publication _, (currently in draft-Sept. 1981).

An important concern for function evaluation is the appropriate level of detail. The recommendation is that basic evaluations be complete (for all applicable control features) down through the functional specification level, as defined in (or appropriate for definition in) the Functional Requirements Document. This notion applies to both controls within the computer and physical/administrative controls external to it (although the latter might not actually be defined in a Functional Requirements Document).

This function evaluation approach is suggested in full realization of the difficulties confronted in determining which functions to include in a Functional Requirements Document, and, when this document is incomplete or non-existent, in examining other sources of application information (e.g., operating procedures, specifications). The reasons are that the functional specification level (1) is a legitimate, commonly-used level, and (2) represents a complete picture of security functions and services with respect to the environment surrounding the application. Completeness is necessary to ensure that major problem areas are not overlooked.

3. Control Implementation Determination: The fact that functions are described in a document or discussed in an interview does not prove that they have been implemented. The existence of most physical and administrative controls can be determined via visual inspection. For controls internal to the computer, testing is needed. In many cases, a short operational demonstration suffices. For example, the existence of a password function can be determined by attempting to use the application and verifying that a valid password is required. Black box (external) testing is generally sufficient for control implementation determination.

4. Methodology Review: It is desirable to gain some assurance that controls are acceptably implemented. The best way to do this without becoming immersed in extensive testing or detailed analysis is to examine the methodology used to develop the application. This step applies regardless of whether the application is currently under development or has long been operational.

Methodology review contributes to a confidence judgment on the extent to which controls are reliably implemented and on the susceptibility of the application to flaws. If review findings suggest that the implementation cannot be relied upon, detailed evaluation is required in order to find specific flaws. Specific flaws are far preferable as certification evidence than a general judgment of low confidence.

Areas of concern in reviewing an application development methodology for certification are summarized below. Several of the areas also apply to security products obtained from vendors.

1. Documentation. Is there current, complete, and acceptable quality documentation? This applies to both developmental and operational documentation.

2. Objectives. Was security explicitly stated and treated as an objective? Were security requirements defined?

3. Project Control. Was development well controlled? Were independent reviews and testing performed and did they consider security? Was an effective change control program used?

4. Tools and Techniques. Were structured design techniques used (e.g., modularization, formal specifications)? Were established programming practices and standards used (e.g., high order languages, structured walk-throughs)?

5. Resources. How experienced were the people involved? What were the sensitivity levels or clearances associated with their positions?

2.1.4 Detailed Evaluation (Sec. 2.4)

In many cases a basic evaluation does not provide sufficient evidence for certification. Examples are cases where (1) basic evaluation reveals problems that require further analysis, (2) the application has a high degree of sensitivity, or (3) primary safeguards are embodied in detailed internal functions that are not visible or suitable for examination at the basic evaluation level. These situations require detailed evaluations to obtain additional evidence and increased confidence in evaluation judgments.

Detailed evaluations analyze the quality of security safeguards. Primary tasks are the examination of the application from three possible points of view and the use of any of several techniques for detailed focusing:

1. Functional Operation (Do controls function properly?)
2. Performance (Do controls satisfy performance criteria?)
3. Penetration Resistance (How readily can controls be broken or circumvented?)
4. Detailed Focusing (What security components need detailed analysis? What really happens in the detailed processing of a transaction?)

The tasks in detailed evaluation are performed as needed. Testing is the most common technique used. Other validation and verification techniques are also available and are becoming more widely used.

1. Functional Operation. This point of view is most often emphasized, since it assesses protection against human error and casual attempts at abuse. Tests of functional operation examine areas

BEST COPY AVAILABLE

5. Removing remote access (thus relying more on physical security).
6. Granting conditional accreditation for a 'shakedown' period before added trust is granted.

2.3 RECERTIFICATION AND REACCREDITATION (SEC. 2.7)

Once an application has been initially certified (whether during development or after becoming operational), the work is not over. As an application or its security environment changes, recertification and reaccreditation are needed.

It is not practical for the accreditor(s), who might be a senior official or a committee of officials, to personally approve every change. On the other hand, substantive changes do require reaccreditation. This gives rise to a need for levels of recertification and reaccreditation based on levels of change. The three levels suggested are: (1) major, affecting the basic security design; (2) intermediate, affecting two or more security software modules or a major hardware component; and (3) minor, within one security software module. The elements of the certification process and the organizational placement of the accreditor differ for each level, with more extensive changes requiring both more extensive evaluation and higher placement of reaccreditation responsibility.

Change control (configuration management) can provide an important assist to recertification and reaccreditation since it is required during both development and operation. Every change should be reviewed for its impact on prior certification evidence.

2.4 [redacted] SEC. 1.5)

For certification and accreditation to be used properly, it is essential to know what evaluation techniques are available and when to apply them. There are four groups of techniques currently used in security evaluation that can be used for certification, either alone or in combination. They differ from one another in their purpose and/or in the organizational entities that use them. [redacted] (See Figure 2 for their relation to life cycle phases.)

[redacted]

[redacted] It can also be used to determine how many resources to budget for security and where to allocate these resources. It can be performed at the such as control operation, parameter checking, common error conditions, control monitoring, and control management. Software tools for program analysis and formal verification methods are applicable.

2. **Performance:** There is much more to the quality of safeguards than proper functional operation. Performance factors relevant to security include availability, survivability, accuracy, response time, and throughput. Stress testing is a useful evaluation technique.

3. **Penetration Resistance:** Penetration resistance evaluation can be used to establish confidence in security safeguards. It can also find and correct flaws, although recent history has shown the inadequacy of 'find and fix' as an approach for achieving security. Since penetration resistance evaluation is different in kind from other forms of evaluation, it can at times play a useful role in certification.

4. **Detailed Focusing:** It is rarely feasible or desirable to examine everything in detail. In addition to evaluation from some or all of the points of view discussed above, two other strategies are especially useful for focusing on narrow portions of the security picture: one based on security components and one based on situational analysis.

Security-relevant components upon which attention might be focused are assets, exposures, threats, and controls. Examples of possible types of analysis include asset value, asset exploitation, exposure impact, perpetrator, control, work-factor, and safeguard tradeoff. It is difficult to anticipate precise needs for such studies when planning certification, although it is safe to assume that some will be needed.

Situational analysis addresses the problem of application complexity, which limits not only the percentage of the application that can be examined, but also the degree of understanding attainable for those portions that are examined. Two useful forms of it are (1) the analysis of attack scenarios and (2) the analysis of transaction flows. Both can be used to complement the high-level completeness of basic evaluation by providing detailed, well-understood examples.

2.1.5 Report Of Findings (Sec. 2.5)

The security evaluation report is the primary product of certification. It contains technical and management security recommendations and is the primary basis for the accreditation decision. The report should:

1. Summarize the security standards or policies that were applied.

2. Summarize the controls that are in place.
3. Summarize major vulnerabilities, recommending which should be corrected and which should be left as residual.
4. Recommend and prioritize corrective actions, if warranted, along with anticipated costs and impacts. Recommend operational restrictions where necessary.
5. Summarize the certification process, so the accreditor(s) can determine how much confidence to place in the findings.
6. Include a proposed accreditation statement (which might be positive or negative).

The certification process should produce the security evaluation report plus other documentation that can be used to support the findings and to evaluate the certification process itself.

2.2 ACCREDITATION (SEC. 2.6)

The accreditor(s) is(are) responsible for evaluating the certification evidence, deciding on the acceptibility of application security safeguards, approving corrective actions, ensuring that corrective actions are accomplished, and issuing the accreditation statement. Aids to be used to assist in this process include answers to questions on resources used (how much, who), processes used (review mechanisms, coordination of findings), and report content (reasonableness, support of findings). Accreditation responsibilities must be integrated into the normal decision-making process of the organization.

Since applications that warrant certification and accreditation are usually important to organization operations, most flaws will not be severe enough to remove an operational application from service. There are many intermediate accreditation alternatives available. The most common is to withhold accreditation pending completion of corrections. Many types of operational restrictions are also possible. For example:

1. Adding procedural security controls. Restricting use of the application to sites that have compensating controls.
2. Restricting the application to process only nonsensitive or minimally sensitive data.
3. Removing especially vulnerable application functions or components. In a network environment a particularly weak node might be excluded from the network.
4. Restricting users to only those with approved access to all data being processed or to those who have passed a background investigation. Restricting use of the application to non-critical situations where errors or failures are less severe.

beginning of the application life cycle and, with user inputs and policy requirements, can provide the basis for application security requirements. When performed later in the application life cycle, it is useful for evaluating security when reliable data exist on threats (e.g., occurrences of fires and floods). Under these conditions, the evaluation may be used for security certification. Risk analysis is usually performed under the direction of people internal to the application in question. (Sec. 1.5.1)

2.4.2 [redacted]

VV&T is a process of review, analysis, and testing that should be performed on an application throughout its life cycle but is particularly cost effective when performed during the early life cycle. Validation determines the correctness of the application with respect to its requirements; verification checks the internal consistency and completeness of the application as it evolves and passes through different levels of specification; and testing, either automated or manual, examines application behavior by exercising it on sample data sets. The performance of VV&T provides a powerful quality assurance technique for applications, and when requirements include security, VV&T becomes an important evaluation technique for security certification. [redacted] however, for critical applications it may be done by an independent body. (Sec. 1.5.2)

2.4.3 [redacted]

Security safeguard evaluation methods are primarily concerned with assessing the security solution. They can be thought of as a specialized form of VV&T. They involve validating security requirements, examining safeguards, and determining whether safeguards satisfy security requirements. Numerous methods are being used for this type evaluation (i.e., checklists, control matrices, weighted ratings for levels of security produced by controls). [redacted] (Sec. 1.5.3)

2.4.4 EDP AUDIT

EDP audit, a subdivision of internal audit, assesses controls against control objectives in agency applications that rely on computers. It is usually broader in scope than just the consideration of security issues. Like security safeguard evaluation, it assesses compliance with policies, existence of controls, and adequacy of controls, but EDP audit might also address cost and efficiency in meeting mission objectives. When control objectives for security are considered, EDP audit becomes a form of certification evaluation. Unlike security safeguard evaluation, however, EDP audit is an activity external to the organizational division in which the application resides and is used by higher-level managers to manage the organization. (Sec. 1.5.4)

Life Cycle Phase	Security Concern	Preferred Security Process to be Applied
I. INITIATION	A. Understand the security problem: identify security risks; determine their magnitude; identify areas where safeguards are needed.	Risk Analysis
	B. Define security requirements.	
II. DEVELOPMENT DEFINITION DESIGN PROGRAMMING TESTING	A. Validate security requirements.	Risk Analysis, VV&T
	B. Assess recommended and implemented safeguards; determine whether they satisfy requirements.	Certification
	C. Approve for operation.	Accreditation
III. OPERATION AND MAINTENANCE	A. Reassess security risks.	Risk Analysis Safeguard Eval. EDP Audit
	B. Reassess safeguards.	Recertification*
	C. Approve for continued operation.	Reaccreditation

*If risk analysis, VV&T, certification and accreditation were not performed during development, they might be performed initially during operation. It is far preferable to perform them during development, however.

Figure 2. Life Cycle Phases and Security Processes

SECTION 3

ESTABLISHING A PROGRAM (SEC. 1 AND 3)

In order to establish a certification and accreditation program in an organization, five issues must be considered in addition to the issue of application sensitivity.

1. Policy and procedures authorizing and defining the program,
2. Roles and responsibilities defining who does what,
3. Organization structure concerns that influence the program,
4. Scheduling when things are done,
5. Staffing, training and support needs.

3.1 POLICIES AND PROCEDURES (SEC. 3.1)

Policies and procedures should be incorporated in (1) a Program Directive that establishes official authority for the program, issuing from the Senior Executive Officer and (2) a Program Manual defining the processes involved, issuing from the Certification Program Manager. The Program Directive could be part of the directive establishing the overall agency security program and should contain a program summary (including purpose) as well as the delegation of major responsibilities. The Program Manual should reflect the responsibilities of the Certification Program Manager and could use the Guideline (FIPS PUB 102) structure as the basis for such a manual.

3.2 ROLES AND RESPONSIBILITIES

The most important consideration in defining responsibilities is proper selection of the accrediting authority (Accrediting Official(s), Sec. 1.3.1). This might be a high-level manager or a group of officials who are responsible for the system and who have authority to remedy deficiencies. Certification personnel are primarily technical (Security Evaluator, Sec. 1.3.4), although every certification effort requires a manager (Application Certification Manager, Sec. 1.3.3). Also, the organization might have a central coordinator or director for all certification activities (Certification Program Manager, Sec. 1.3.2). In some cases several certification efforts are performed in support of one accreditation decision. It is preferable to integrate such multiple technical certification findings into one final report.

3.3 ORGANIZATION STRUCTURE CONCERNS (SEC. 3.2)

Although there is no universally applicable way to structure the organization of a certification and accreditation program, there are two universal concerns in this area: (1) the need for an appropriate high-level manager as Accrediting Official and (2) the need for Security Evaluators who are as objective as possible. (A sample organization structure is shown in Appendix G.)

3.4 SCHEDULING (SEC. 1.4)

The next issue is when this combined certification and accreditation process is performed. It begins with requirements definition and continues through development, operation, and maintenance of an application. It must be integrated into the life cycle management process. Also, it is far preferable to initially certify and accredit an application under development than after it has become operational. The primary reason is that an application under development is easier to change. A second reason is that certification during development permits the development process itself to be improved.

3.5 STAFFING, TRAINING, AND SUPPORT (SEC. 3.3)

In order for the staffing, training, and support for certifications to be adequate, there is a serious need for management wihin agencies to consider the importance of computer security in general to the proper operation of their computer applications. Sufficient attention must be given to both career paths for security staff and the proper training of such persons in security evaluation. Funding commensurate with the sensitivity of the applications involved should be allocated for such staffing and training as well as for general administrative support.

Two further points are noteworthy concerning technical support. First, it is often best to use both independent and internal people for security evaluation. Independent people provide necessary objectivity, although they are costly since outsiders must take the time to learn details of the application. Internal people are typically less costly, and can benefit greatly from the computer security training and increased security awareness they gain from participation, but are usually less objective. The second point is that certification can make much use of validation, verification, and testing findings and of reviews that are routinely performed during development and operation. It is not practical for certification to duplicate these activities. On the other hand, it is desirable for certification needs to influence them, (e.g., by anticipating and recording these needs in VV&T planning).

Background + INFORMATION
- System development life cycle phases
- roles and responsibilitys of personnel.
  - Need for Top Mgmt oversite -- pg III-18.
  - Validation staff ✓ III-20
  - Internal Audit-III-20

SYSTEM INFORMATION NEEDS ANALYSIS

Key information must be gathered and analyzed with respect to whether the system produces sufficient information to meet user needs. The information gathered relates to user requirements; system and product usefulness and necessity; system timeliness; user satisfaction with system performance and computer-produced output; and value and cost of processing.

The accuracy and effectiveness of controls, as well as system economy, efficiency, and effectiveness begin with the methods used during the system design, development, and modification process, commonly called system development life cycle (SDLC). Furthermore, the SDLC becomes the mechanism by which user needs are designed into the system.

The examination as to whether a system meets the user information needs involves an evaluation of the integrity of the system's design, development, and modification process. In other words, it verifies the integrity of the process that was used to design and develop the system, including the modification process used to make changes to the system.

An evaluation of a system's design, development, and modification integrity includes:

--Understanding the system's design, development, and modification processes, and evaluating these against generally accepted design, development, and modification processes.

--Identifying the roles and responsibilities of persons working on the project.

--Interviewing persons responsible for key system design, development, and modification activities.

--Identifying internal control and audit trail techniques included in the system design.

--Analyzing the results of the system testing and certification processes.

--Identifying mechanisms for controlling systems modifications.

--Reviewing system and program documentation to determine the extent of compliance with NBS guidelines and generally accepted system design, development, and modification processes.

Source: Systems Development Training Material For GAO Evaluators

This evaluation should

(1) Provide background for a review of internal controls over system design, development, and maintenance, and

(2) Identify risks associated with

--the integrity of the system's design, development, and maintenance, and
--the system's achievement of user needs,

both of which establish the level of review to be performed under system economy, efficiency, and effectiveness testing.

The survey steps include:

- --evaluating the SDLC process,
- --evaluating the roles and responsibilities of project personnel,
- --analyzing compliance with generally accepted documentation standards and guidelines,
- --preparing an inventory of system users,
- --analyzing user satisfaction with the system performance, and
- --analyzing user satisfaction with computer output.

SYSTEM DESIGN, DEVELOPMENT, AND MODIFICATION PROCESS

[redacted] Proper controls over these processes help make sure that systems are built to meet user requirements, are developed economically, are thoroughly documented and tested, and contain appropriate internal controls and audit trails. To accomplish these goals, [redacted]

[redacted]

Use of structured approach.

[redacted] The approach must provide review points enabling management to continually monitor and assess progress, performance, and budget status and, where necessary, reevaluate, reschedule, or terminate development work.

The SDLC concept is the recommended management control technique to divide the ADP system development process for significant applications into distinct phases so that management can review the process at appropriate decision points.

The characteristics of the SDLC approach may vary, depending on such circumstances as organizational differences or management philosophy. See table 1 on the next page for a comparison of the SDLC phases defined by FIPS 38 and various other examples of the systems development process. Regardless of the unique definitions, the SDLC approach should encompass periodic reporting to management and surfacing, analysis, and resolution of problems encountered. The technique is as applicable during initial system design as it is during the modification process; thus, it should be used whenever changes are made to a system. Additionally, SDLC is particularly advantageous because it promotes responsible communications among programmers and systems analysts, system validators, users, auditors, and management.

Exhibit 1 separates the SDLC into three major phases--initiation and definition, physical design and development, and implementation and operation--each encompassing various activities or steps. Each step must be completed before the next can be started. At the completion of each step, all previous work is reviewed, and a "go/no go" decision is made. This progression through the phases and appropriate steps thus provides a structured approach to the development process. The three phases may be generally described as follows:

Initiation and definition phase--The user, ideally aided by a systems analyst, issues a formal project request which defines the needs and objectives expected to be achieved from the proposed project. One or more conceptual system design alternative(s) is/are then developed, and each is then evaluated to assess technical and operational feasibility. A cost-benefit analysis is performed to make sure that the selected alternative will produce the desired results economically (a risk analysis is an integral part of the cost-benefit analysis). The planning stage finally results in the completion of a functional requirements document, or project master plan, which summarizes requirements and objectives, assigns areas of responsibility, and schedules the various project activities.

Physical design and development phase--The detailed system design is created, and used to prepare required computer programs. Once the programs have been completed, each program, and then the entire system is tested to make sure that the system is fully operational. Later, processing procedures and documentation--both manual and automated--are prepared for implementation of the system (documentation should be developed using NBS' Federal Information Processing Standards (FIPS) as minimum requirements).

Implementation and operation phase--This phases begins with preparation of needed procedures for converting to newly designed programs and for building new files. This phase also

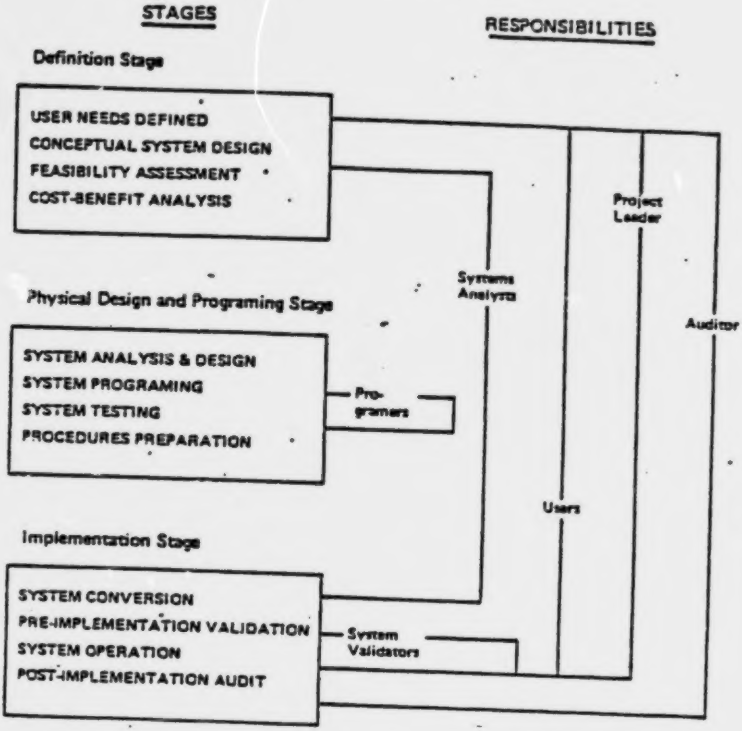
Table 1

Terms Used by Several Organizations (as compared to FIPS '38) to Describe Phases of Systems Development

FIPS PUB 38	INITIATION	ANALYSIS	DESIGN	DEVELOPMENT	IMPLEMENTATION	OPERATIONAL
ARTHUR YOUNG	DEFINITION	ANALYSIS	DESIGN	CONSTRUCTION	INSTALLATION	MAINTENANCE REVIEW
McGRAW HILL	PROBLEM IDENTIFICATION	SYSTEMS ANALYSIS	(A) EXTERNAL DESIGN (B) INTERNAL DESIGN	CONSTRUCTION	IMPLEMENTATION	EVALUATION
VA (5DM/70)	INITIATION	GENERAL DESIGN	(A) FUNCTIONAL DESIGN (B) TECHNICAL DESIGN	(A) DEVELOPMENT (B) TEST	INSTALLATION	POST-IMPLEMENTATION
PUBLIC ACCTG.	SYSTEMS PLANNING	SYSTEM REQUIREMENTS		DEVELOPMENT	IMPLEMENTATION	MAINTENANCE
LARGE GAS CO.	(A) PROJECT INVESTIGATION (B) PROJECT PLAN	CONCEPTUAL DESIGN	DETAIL DESIGN	PROGRAM DESIGN	IMPLEMENTATION	POST-IMPLEMENTATION
LARGE MFG. CO.	PLANNING	ANALYSIS	DESIGN	IMPLEMENTATION		SUPPORT

Exhibit 1

THE SYSTEM DEVELOPMENT LIFE CYCLE


STAGES
RESPONSIBILITIES
Definition Stage
USER NEEDS DEFINED
CONCEPTUAL SYSTEM DESIGN
FEASIBILITY ASSESSMENT
COST-BENEFIT ANALYSIS
Project Leader
Systems Analysts
Auditor
Physical Design and Programing Stage
SYSTEM ANALYSIS & DESIGN
SYSTEM PROGRAMING
SYSTEM TESTING
PROCEDURES PREPARATION
Programers
Users
Implementation Stage
SYSTEM CONVERSION
PRE-IMPLEMENTATION VALIDATION
SYSTEM OPERATION
POST-IMPLEMENTATION AUDIT
System Validators

[redacted]

Once the system has been certified and has satisfied the user acceptance criteria, it is placed into operation. During this phase, the entire system (both manual and automated processes) should be reviewed periodically by the users to ensure that it continues to meet their requirements and by the internal audit staff to ensure that the system (1) maintains the necessary internal controls to consistently produce reliable results, and (2) operates in accordance with agency and federal standards and approved design specifications. These reviews can also be used to determine when the system no longer meets user needs so its replacement or discontinuance can be recommended.

In addition, to properly serve the user during this stage, there must be in place a formal system of control whereby unforeseen problems or changing user requirements can be addressed. At a minimum, such a system should provide for (a) formal, documented change requests and approval, (b) thorough testing, (c) validation and re-certification, and (d) user acceptance.

[redacted]

In addition to providing decision points for the various activities in the system development process, SDLC defines specific responsibilities for the various personnel involved in the process. These responsibilities and associated duties are established at the outset of the project and provide for performance and management accountability. Management control of the project is ensured by formalizing responsibilities as discussed below.

System users--Users must actively participate in the development process to ensure satisfaction of their requirements. Users should be responsible for identifying their needs and submitting them to top management for decisions regarding priorities, resource commitments, and implementation schedules. Users and managers should work together to define functional requirements for ADP systems and develop plans to economically meet user needs. All functional users should participate in the design and development process to ensure agreement on objectives, specify information needs, and outline the scope of the development.

Users must be involved throughout development to ensure that a project is on schedule, within the budget, and meets service goals. Users should review a project at significant milestones to verify that all requirements are being met and that resource expenditures should continue. Users should be fully informed of progress and be knowledgeable enough to approve completed work and continuation of the project. Through participation and communication with ADP managers, users have greater assurance of a reliable and effective system and are better prepared to train their personnel in using the system. The users' participation also contributes to greater management control of development resource expenditure.

To ensure that the user has been fully informed and involved, there should at a minimum be a user sign-off evidencing acceptance of or concurrence in (a) each deliverable provided or milestone met, (b) changes in the project management or planning, and (c) major modifications to the system design, cost, or operation.

Project Steering Committees--To assure top management control during the entire course of development, a management group, commonly called a project steering committee, should be formed to regularly oversee and review progress and make decisions at each critical stage. It is particularly important to have such a committee if the system is intended to serve the needs of two or more distinct user organizations.

The committee should decide whether to initiate, continue, revise, or [illegible] and should consider any strategic matters [illegible] the project, provide overall direction, and establish both accountability and primary management controls. The project steering committee should meet regularly to review and analyze progress and performance and approve completed work and plans for the next phase before the project proceeds.

The active membership of the project steering committee should include representatives from the user organization, operations, system design and development, security, and the top managers having overall responsibility for coordination and control of ADP resources. The agency internal audit staff should be involved in an advisory or monitor capacity, and, depending upon the particular system in question, representatives from other areas (e.g., law, personnel, finance) might be involved. All committee members should be at a sufficiently high level to make authoritative decisions based on priorities and directives established by the agency head or the executive ADP management committee. For this reason, the project steering committee should have a formal charter defining the committee's authority and accountability for system development.

Project Manager--Each major ADP system should have a full-time project manager assigned as a central authority to provide daily direction, coordination, and control. The project manager should generally represent the principal end user and have a formal charter defining the manager's specific duties and authority.

The project manager, subject to the approval of a project steering committee, should have authority to decide on personnel allocations, establish project plans, schedules, and budgets, and to conduct most technical activities. This person should coordinate the related functions in system development and lead and direct the project team. As a leader of a team composed of persons with mixed skills, the manager must provide a well-defined and structured environment within which system development can progress in an orderly manner. The project manager must also serve as the interface between users, programmers, analysts, system validators, and top management. Also, the manger is the key person in assessing and negotiating trade-offs and arranging meetings with the project steering committee to keep it informed of project status, obtain required approvals, and refer problems which are outside the manager's authority. Such problems usually relate to conflicting priorities, unmet resource requirements, schedule slippages, or events requiring a major change in direction. If the system is being developed by a contractor, close coordination between the contractor and agency project manager is essential.

The project manager should ensure that government and agency regulations are followed, federal standards are applied, and that system requirements are met.

Systems analysts--Systems analysts are responsible for translating user needs into a conceptual system design. This conceptual system is then used by systems analysts and users to make sure that the system is technically and operationally feasible and can produce the desired results at a minimum cost. After design is completed the systems analysts

--create detailed system design specifications used for detailed programming,
--review program documentation, after programming has been completed, to make sure that the detailed design specifications have been followed,
--review program test procedures and results to make sure that each program has been thoroughly tested and that each program and the entire system is fully operational,
--complete preparation of accurate system documentation, including instructions for users, control personnel, and operating personnel, and
--prepare procedures and documentation for system conversion so that the system can be validated and, once validated, begin operation.

Programmers--Programmers are responsible for preparing computer programs in accordance with systems analyst design specifications. After programming is completed, the programmer tests each program to make sure that it is fully operational and documents each program and the types of tests performed. This documentation should be accurate, complete, and up to date because it is critically important to persons responsible for all other phases of the system development life cycle.

System Validation Sta[illegible] validators or acceptance testers are resp[illegible]forming comprehensive pre-implementation tests of all new systems and all modifications to existing sy[illegible]. This tes[illegible]uates whether the entire system--both the manual and [illegible]--is performing and functioning as specified by the user and as designed by the systems analyst. No system should be placed in operation without the system validators' written certification that the entire system performs in accordance with all functional and performance specifications. For these reasons, the system validators must control all planned system changes so that no changes can be made to any program without their prior approval. Emergency modifications to existing systems must also be so validated, although conditions may necessitate such validation being performed after the fact. Thus, all changes, both planned and emergency, are subject to review and approval by system validators.

However, in order to fulfill its independent role the system validation function should not be directly responsible to the project manager during system development or to user or systems management during the operational stage.

Internal Auditor--Internal auditors are responsible for assuring top management that all systems operate in conformance with management [illegible] design specifications and also include internal controls needed to produce consistently reliable results. The internal auditor should also act as a user to make sure that all systems contain necessary audit trails. To accomplish these tasks, internal audit must actively participate in the system development process by reviewing each step as it is completed. Then, once the system is operational, the auditor has a continuing responsibility to review both general and application controls to assure that the system continues to perform in accordance with management policy and produce consistently reliable results.

However, despite certain functional similarities, the role of the internal auditor is to be carefully distinguished from that of the system validator. The latter is responsible for detailed testing and certification of all systems and changes thereto prior to implementation. As such, it is a production-oriented quality assurance function and is not the proper role for the independent internal auditor.

Documentation

Documentation is the process of describing on paper how a computer-based system operates. The objective of good documentation is to provide a clear, understandable description of the system. Good documentation increases the ease and accuracy of computer program maintenance and provides the basis for evaluating internal controls in the system. Documentation is used to:

- --Provide management with a clear understanding of systems objectives, concepts, and output and to ensure that their policies are adhered to;
- --Ensures correct and efficient processing within both data processing and user areas;
- --Provide a convenient reference for systems analysts and programmers responsible for maintaining existing systems and programs;
- --It provides a primary basis for training; and
- --Serves as a basis for review of accounting and internal controls by internal and external auditors.

Types of documentation that should be developed are included in FIPS Publication No. 64, "Guidelines for Documentation of Computer Programs and Automated Data Systems for the Initiation Phase" and FIPS Publication No. 38, "Guidelines for Documentation of Computer Programs and Automated Data Systems," prepared by the National Bureau of Standards, Department of Commerce. This documentation includes the following:

- --Project request document. Provides the means for a user organization to request the development, procurement, or modification of software or other ADP-related services.
- --Feasibility study document. Provides an analysis of the objectives, requirements, and system concepts; an evaluation of alternative approaches; and identification of a proposed approach.
- --Cost-benefit analysis document. Provides managers, users, designers, and auditors with adequate cost-benefit information to analyze and evaluate alternative approaches.
- --Functional requirements document. Provides a basis of mutual understanding between users and designers of the computer-based system including system requirements, operating environment, and development plan.
- --Data requirements document. Provides data description and technical information about data collection requirements.
- --System specifications. Details the requirements, operating environment, and design characteristics for a system or subsystem.
- --Program specifications. Details the requirements, operating environment, and design characteristics of individual computer programs.

--Data base specifications. Details the identification, logical characteristics, and physical characteristics of a particular data base.
--Users manual. Describes fucntions performed by the computer system in non-ADP terminology and serves as a reference document for preparing input data and parameters and for interpreting system outputs.
--Program maintenance manual. Provides information necessary to understand system's computer programs, operating environment, and maintenance procedures.
--Test plan. Provides a strategy for testing the computer-based system, including detailed specifications, descriptions, and procedures for all tests, and test data reduction, and evaluation criteria.
--Test analysis report. Documents tests results and findings, presents demonstrated capabilities and deficiencies for review, and provides the basis for preparing a statement of system readiness for implementation.

The act of preparing documentation helps prevent incomplete economic evaluations or inadequate specifications. Documentation facilitates communications and subsequent system modifications and may act as a deterrent to fraudulent manipulations of systems. Documentation provides the means for effective system review by management, users, technical personnel, and auditors. It is essential for correcting design errors and for system maintenance.

AUDIT PROCEDURES

1. Use: --Summary of the System Design, Development, and Modification Process - SD-1,
   --Principal Users Inventory - USR-1A,
   --Evaluation Guide: User Satisfaction of Information Needs - USR-1B, and
   --Evaluation Guide: User Satisfaction of Computer Output - USR-1C.

   To document:
   --the existence and use of a SDLC process,
   --SDLC techniques actually in use, as identified by interview and limited observation,
   --degree of achievement of satisfying user needs through the use of SDLC, and
   --inefficiencies of the SDLC process and risks in meeting user information needs.

2. Summarize the overall SDLC environment, the satisfaction of user needs, the major SDLC process weaknesses, and associated risks or inefficiencies. This summary should be included as a major section of the system profile.

Chapter 12

APPLICATION SYSTEM DEVELOPMENT CONTROLS

The adequacy and effectiveness of controls included in computer application systems are affected by the methods and procedures used during the system development process. Controls over the system development process are important for three reasons. First, good development controls assist in managing costs and schedules. Second, they help ensure that appropriate application controls are built into application systems being developed. Third, they ensure that application controls are properly tested before application systems become operational. By carefully controlling the system development process, one can achieve higher levels of accuracy and reliability in the computer application systems developed and satisfy the goals of developing quality application systems within cost and on schedule.

Elements of the application system development process documented by SRI include project management, programming techniques, development and acceptance testing, program change control, documentation, and data base administration. Each of these elements is discussed in this chapter. Specific techniques and controls within each element identified during the study are presented within the relevant sections of the chapter.

This chapter presents a list of controls and techniques currently being used by organizations during the application system development process to develop more reliable and accurate systems within cost and on schedule. Table 12-1 presents a list of all controls and techniques, organized by control area and by control type within area.

The first technique presented is the systems development life cycle (SDLC). This is presented first because it provides a structure for the application system development environment that encompasses all the above-mentioned elements of the application system development process. All of the other elements mentioned above will be discussed separately in the logical order in which they occur within the SDLC.

A System Development Life Cycle Example. The SDLC is a technique used to divide the system development process into a small number of distinct phases with formal management control points placed between and during each phase. The objectives in using an SDLC technique are twofold: to provide a more structured management scheme for controlling costs and schedules, and to ensure proper and responsive communications channels among users, EDP auditors, hardware planning personnel, top management, and the data processing personnel responsible for developing the application systems.

The majority of companies interviewed which were using an SDLC technique did so to ensure that major application systems were developed on schedule, within cost, and to the user's satisfaction. In addition, interviews indicated that federal and state legislation relating to controls over personal privacy, electronic funds transfer, and other such activities was becoming a major concern of top management. As this legislation specifies certain legal liabilities that, if incurred, could seriously damage the entire organization permanently, top management is seeking more assurance that computer applications systems are designed, implemented, tested, and operated in the best possible ways. Many organizations felt that a properly specified and controlled SDLC was the best technique available today to accomplish this task.

The following representative example of how one large government organization has defined and uses the SDLC serves as a basis for discussion of the general controls and techniques identified in use during the study. However, other organizations define and use slightly different SDLC techniques, and such differences are discussed after the example. The government organization defines the SDLC as follows:

- Project definition – That phase whose primary purpose is to define the user requirements and uses for the system.
- System analysis and design – That phase in which an overall description of the system is prepared.
- Detailed design and programming – That phase that focuses on the internal components of the system and the development of computer programs needed to form the system.
- System test – That phase in which the system is exercised to determine the correctness and com-

Table 12-1
APPLICATION SYSTEM DEVELOPMENT CONTROL STRUCTURE

Control Area	Control
System development life cycle	Review user requirements
	Review project organization
	Review hardware requirements
	Review internal controls
	Require user or internal auditor sign-off
	Review detailed design documents
	Review file requirements
	Review costs and schedules
	Review test plan
	Review user and operational documentation
	Review test results
	Require user and internal audit sign-off
	Review conversion plan
	Review adequacy of documentation
	Identify user problem areas
	Identify system development problems
Project management	PERT control technique
Structured programming	Audit trail
Acceptance testing	Comparison of base case data to system produced data
	Manual review of output reports
Program change control	Require formal written request
	Review all changes
	Use of change control committee
	Restrict number and type of persons who make changes
	Require operator and programmer to make changes
	Modify SMF to obtain reports of all changes to load libraries
	Require report of quick fixes
	Limit number of times changes are made per time period
	Use program packages to control access to source libraries
Documentation	Provide flow of all application system data flow
	Specify how the programs implement controls
	Specify how programs are to be operated, backed up, and recovered
	Maintain up-to-date changes to accepted documents
	Specify allowable user commands and functions
Data base administration	Develop standards
	Establish and monitor standards
	Document and provide procedures to control operations
	Measure effectiveness of performance and integrity controls
	Develop security/control education programs

pleteness of implementation to the user requirements as manifest in the design documents.

- Conversion – That phase in which the tested system and operational procedures are initiated to move the system into a full operational mode.

Other organizations include one or two additional phases broadly defined as:

- Operational – That phase concerned with ongoing operation, program changes, and maintenance
- Postimplementation – That phase concerned with whether the system performs as intended by the user, with focus on methods to improve the development process.

Figure 12-1 shows details of the five phases of the development cycle and the 15 control points used by that government organization. During the development of application systems, all requirements specified at each control point must be satisfied before that phase or the next phase can be continued.

Basically two types of control points are used in this SDLC: those affecting the quality of systems being developed from a computer processing point of view and those used to interface to users and others outside the data processing department. System quality control takes place primarily at Points 1 and 4-13.

- Control Point 1 – The EDP auditor, user, and the project leader review the project organization, the arrangements with the user for communications, and the plans and work program for the design. This central point helps the project leader to establish a good working relationship with the user to ensure that the system reflects user requirements.

- Control Point 4 – The user, EDP auditor, and project leader review the project organization resulting from the first phase, the communication links established between team members, users, and EDP auditors, schedules and work plans, and other items germane to the specific project.

- Control Point 5 – The EDP auditor, user, project leader, and design analysts review the detailed design output reports for completeness and clarity. The EDP auditor attempts to ensure that sufficient design documentation exists to allow for a clear understanding by the test team and the EDP audit staff.

- Control Point 6 – The user, EDP auditor, project leader, and design analysts review the file requirement specifications and the input requirements associated with them. The user attempts to ensure that the file requirements do not implicitly or explicitly change the original system specifications.

- Control Point 7 – The user, project leader, EDP auditor and other data processing personnel responsible for hardware planning review the equipment requirements to meet the requirements of the designed systems. Completeness of the equipment requirement is important to avoid unanticipated equipment costs at a later date.
- Control Point 8 – The EDP auditor, user, and project leader review the design from cost, data processing standards, and general management points of view. The project leader is interested in ensuring that all loose ends from the past two phases are in place before moving into the detailed design phase.
- Control Point 9 – The EDP auditor, project leader, user, and data processing personnel make a final review of plans, equipment, costs, project organization, and communications channels to ensure that all participants have agreed upon the status and direction of the project. The project leader is primarily concerned with assuring top management that sufficient systems analysis and design have taken place before the detailed design phase.
- Control Point 10 – The EDP auditor, user, and project leader review the documentation scheme and documents available describing the file systems, interface data handler programs, and program run documents for compliance to standards, completeness, accuracy, and clarity. The project leader is primarily concerned with ensuring that the project team is providing adequate documentation to meet data processing and user documentation standards.
- Control Point 11 – The EDP auditor, the project team members, testers, and user review the detailed system design to ensure that it follows from the general system design and still meets the user's requirements. In addition, the test plan is reviewed for completeness, timing, and cost. The conversion plan and associated paperwork are reviewed for reasonableness, completeness, and clarity. As this is the last checkpoint before the test phase, the project leader takes special care to ensure that the original design requirements are still intact or that a traceable trail exists that explains to top management and users why the system has changed.
- Control Point 12 – The EDP auditor, testers, user, and project leader review the test team organization to ensure that the proper people are present and that the project test plan is complete and consistent. The project leader is primarily concerned with assuring himself that the test plan will completely test the system and in particular will test the internal controls designed in the system.
- Control Point 13 – The EDP auditor, user, and testers develop test data, build masterfiles, review test results, and monitor the test plan progress to ensure that it is adhered to throughout the test phase.

General management and other organizational interface controls occur primarily at Control Points 1, 2, 3, 4, 14, and 15. As Control Points 1 and 4 have been described, only Points 2, 3, 14, and 15 are discussed below.

- Control Point 2 – The user, EDP auditor, and project leader review the analysis and planned cost for completeness and accuracy. In addition, the project control and communications plan is discussed and changed if necessary. The user plays a major role at this control point in assuring himself that proper analysis has taken place.
- Control Point 3 – The user, EDP auditor, and project leader review the conceptual design documentation for accuracy, completeness, and any changes that may have occurred. A revised cost-benefit plan is developed, and the EDP auditor presents the findings to top management.
- Control Point 14 – The user, EDP auditor, computer operation personnel, and the project leader review the conversion plan for completeness of detail and personnel involved. Plans for communicating the production schedule to top management are discussed as well as other miscellaneous considerations germane to the specific project.
- Control Point 15 – The user, EDP auditor, and project leader review all problems not yet resolved, adequacy of documentation, and any incomplete activities identified. Final reports on the project status can then be written by the EDP auditor.

Management at the government organization using the above SDLC told SRI that they would like to include a postimplementation phase in their current cycle. They feel that after the application system has been used for a few months, problems can be identified by the users that will point out methods to improve the application system in question, as well as improve the entire SDLC technique used.

Controls and techniques used in connection with the SDLC by other companies interviewed include:

- Daily reports to the managers of the functional areas during the critical stages of application system development.
- Required sign-off by the internal audit department on all new application systems developed. This control forced the internal auditors to become involved with the project team at an early stage of the SDLC.
- Postimplementation performance reviews conducted by the internal auditors and users to determine whether the operational system functions as expected and has been adequately documented from a user's point of view.

In summary, using some type of SDLC does provide more management structure and an opportunity for users and EDP auditors to participate in all stages of application system development. It does not ensure that users and EDP auditors will do their respective jobs well or that project leaders will communicate as well as everyone would like. SDLC is a good application system [illegible] because it provides [illegible] develop [illegible] and higher quality [illegible] provide [illegible] degree of data integrity. SRI believes that there is a definite trend to the use of some types of SDLC in organizations that are heavily involved in application system development.

2. PROJECT MANAGEMENT

Project management is a control technique that provides a formalized means of measuring progress of application system development through use of reports, such as periodic project status reports. The objective of project management is to provide a well-defined and structured management environment that offers a means of measuring progress during application development. The main elements of project management include:

- A means for establishing intermediate work products that can be approved by users participating in the development process. The SDLC noted above is one technique for providing the required structure and work products.
- A means of providing project status reports that indicate expenditures versus budget as work progresses and actual completion dates versus scheduled dates.
- A means of providing periodic status reports to top management.

Basically, two project management techniques were used for controlling the development process reported during the study: the team approach and the program management group.

Team Approach – The team members included persons from data processing, the user group, the internal auditors group, and the hardware planning group. This team was usually coordinated by the project leader from data processing. Team members worked together throughout the development process, although not all team members were required at all phases. The SDLC example above is an example of this technique.

Program Management Group – This group is usually within the data processing department, whose charter is to monitor application system progress and to ensure that proper controls are included in the application system.

A large company interviewed formed a System Acceptance Group (SAG). The SAG plays a significant role in the early system design stage of the system development process, during the time just before actual operation of the system, and to a lesser degree during acceptance testing. It plays only a passive role in the detail design, conversion, and operations phases. One of the prime purposes of the SAG is to ensure that proper controls are included in the design of the application system. It also maintains up-to-date manuals describing control standards in a form that permits application of the standards in effective ways in an information system design.

The function of the SAG is split between overall control of the system development process and ensuring that controls are included in any system design. As such, the group is more or less continuously involved in the control aspects of systems in both the design and general production use of the system. While administering controls for system development, the group also develops and enforces standards in a form that permits their implementation in an information system.

Regardless of which project management technique was employed, a number of organizations interviewed were using Gantt charts and computer-based PERT control techniques to assist project management in keeping abreast of schedules and costs.

[redacted] here is a rapidly accelerating trend to the use of project management techniques in large and small organizations for all application system development, be it a large or a small project.

STRUCTURED PROGRAMMING

Structured programming is a technique that provides specific guidelines to programmers on how they may use a programming language and how each program fits together to form an application or operating system. The objective in using structured programming techniques is to develop more usable and effective programs. "Usable" implies that the program can be read and understood by technical persons who did not write it, including users and EDP auditors. "Effective" implies that the program is designed to fit into an overall application system scheme so as to reduce redundancy and ensure processing efficiency.

Structured programming is a technique for system builders that renders systems easier to build, maintain, and alter. It is a discipline that is used primarily in the detail design and programming stages of the development process. As such, it uses a stepwise top-down approach, in which program modules are organized by functional specifications into a balanced hierarchical structure with minimum side effects on each module.

Structured programming involves a team approach to detailed design, with team members being used to "walk-through" the design and coding of components. The effect is that the design and code can be viewed by other than the originator to detect faulty logic, hard-to-follow code, and the extent to which the design meets prespecified objectives.

An example of using structured programming as a control over the production of code was reviewed at a major insurance company. This approach is relatively new for this organization, but the success attributed to it has been significant. The main advantage has been that the company finds programs easier to read, correct, test, and modify. It has also facilitated rapid training and made cross-training easier. This organization believes that, in spite of a high turnover rate in the programming staff of a large project, the project was completed on time because structured programming was used. This is perhaps the strongest testimony for this organization's experience with the use of structured programming.

The main areas of structured programming used by this organization include the use of:

- A semistrict program structure allowing GO TO statements in a downward direction within a section domain.
- Fixed column indentation for both procedure division and data division sections; in addition, a maximum of one verb per line and specific columns for operators.
- Structured walk-throughs whereby at least two peers completely trace or walk through the code generated by another programmer.

Another large insurance company has experimented with structured programming techniques on a pilot project. In most cases, the techniques were of significant value and could be used with little conflict with current methods and procedures. The overall results of the pilot project proved that the techniques were effective in increased quality of the system and reductions in time spent to produce the system. In system test, less than 60% of the planned time was spent. This is primarily attributed to the walk-throughs and specification techniques. In terms of programmer productivity, approximately 6% savings were realized over what was budgeted. By far the most useful result of the test was the high quality of the system produced, which, in turn, resulted in greater user acceptance and confidence in the system.

The insurance company attempted the use of many techniques involved with structured programming. In each case, a postproject analysis was performed summarizing the advantages, problems, and recommendations for continued use. The major techniques that were used are described in the following paragraphs.

Team Organization – This technique calls for the establishment of an integrated team consisting of one project leader/analyst, two programmer analysts, one to three programmers, and one programmer librarian. The experience of the team approach was generally good. The difficulties encountered were mainly ones of separation of duties among team members and not specifying the roles each member would take in the specification and programming phases. The approach was recommended for continuance of other projects.

Top-Down Design – This technique consists of designing program logic by specifying higher level functions first and then determining the subfunctions required to implement these higher level functions. The main advantage of this approach is that it forces a thorough analysis before program design begins. The recommendation was enthusiastic for continuing this approach on new projects.

Segmentation – During detail design and programming, it is advantageous to keep programs and modules in the form of routines called segments, with each segment having but one entry and exit. This approach follows naturally from the top-down design. It was also felt useful from the point of view of reducing code redundancy and defining clean interface points for other parts of the system.

Structured Coding – This approach or discipline is used to depict the process of coding whereby there are conventions used for syntax, program format, restricted and controlled branching, and disciplines on logic.

Walk-Throughs – The walk-through consists of a planned review of all system specifications and coding by peers of the developers. The walk-throughs were instrumental in uncovering a majority of errors during the preinstallation and test phases of the system. In adopting this approach, the organization found it more worthwhile to allow others to review the specifications and code before a joint meeting. The review meeting then became mainly a question answering and resolution session.

Top-Down Testing – This approach requires that the skeleton control modules be tested first, then progressing down the module structure until finally the entire system is tested.

Programmer Librarian – The programmer librarian actually serves the purpose of documenting all source codes. This function is responsible for getting codes keypunched, updating the source library, and other general program documentation. The difficulty with this approach is that the librarian function may be unable to be responsive to program changes coming from the coders and, in general, to interface or integrate into the actual coding process. Another more subtle disadvantage is that programmers prefer to do some of the librarian tasks themselves, as a break in the routine of coding. It was recommended that future projects not incur the expense of a librarian, allowing the programmers themselves to be responsible for all librarian functions.

In summary, use of structured programming techniques does provide for more usable and auditable programs. As it is a relatively new concept, there are some problems in changing to it from traditional programming techniques. However, it is a good control technique because it does provide an improvement in the quality of the application system as well as the individual programs that make it up. SRI found a slowly developing trend toward the usage of structure design and programming techniques, primarily in large organizations that develop major application systems.

[redacted]

The type of acceptance testing must be planned for in the early phases of the system development process. The details of acceptance testing can be based on a class of transactions that are defined, in general, during the system definition and design phase. These transactions are used to depict data that is input to the system and output from the system. The design process then takes into account the volumes for each of the transactions and the types of processing that need to be performed between input and output.

The definition of transactions heavily involves the various user departments either supplying or receiving data processed by the systems.

Base Case Testing – An approach to system testing using transactions that is used by one manufacturing organization is called base case testing. Base case testing is a technique to execute computer application systems and/or programs using test data sets developed as part of a comprehensive testing program, and to verify processing accuracy by comparing processing results with predetermined test data results. The base case test consists of test files containing data to test valid and invalid conditions, a predefined set of input and output transactions, and an automatic or manual procedure to compare test results with predefined results.

The scope and composition of a base case are determined by consideration of input transaction types, computer programs, output reports, masterfiles, edits and validations, and system controls. The analysis and procedures for determining correct operation of the system using the base case consist of file comparison programs, program monitoring aids, file dump utilities, comparison of base case data to system produced data, and manual review of output reports.

This organization makes heavy use of representatives from users' departments in forming the base case. In addition, the organization believes that individuals responsible for the base case must be more than part-time participants. Sufficient time is planned for both the definition and use of the base case. The planning for the base case starts in the first phase of system development, the system definition phase. During the next phase of system development, the base case is defined in detail and also serves the purpose of assessing the design as it progresses toward detail design and programming.

The base case approach to testing is used for system test at the preinstallation stage, as well as during the production life of the system. As the system is used, modifications are bound to be made. As these modifications are made, the base case test is modified and used to test the new version of the system to ensure correct operation and avoid the bugs that inevitably result from minor changes.

Base case testing provides comprehensive system verification and compliance testing. However, it does require very substantial resources to maintain extensive data sets after initial installation. Consequently, it will probably be some time before extensive use of this technique is made in most organizations.

Transaction Driver Test – The transaction driver technique is a procedure that uses software to input transactions to an application system as part of the test procedure before the system becomes operational. The approach of testing systems through the use of transactions and an on-line transaction driver is being used by a major oil company. This company uses an on-line test system to exercise its new application systems. This system is used to generate and track transactions through the system being tested. The transactions do not impact the live data bases except for query purposes. In particular, they do not allow updates to the live data base during system test. All transactions from the on-line test system terminal are included on the log tape for comparison against predefined test tapes. The transactions that would normally go to posting on the data base are stripped away and stored on test tapes.

This approach is useful for the testing of on-line systems that are transaction driven. Through the use of this technique, any remote access to the system can be tested before actual production use. The approach is not directly applicable to nontransaction systems or to systems that are not on-line. In addition, the approach would be less effective in testing systems where there is a high incidence of file updating. However, for on-line transaction driven systems, the approach is extremely effective.

During the field interviews, the following general concepts of acceptance testing were identified:

- Acceptance test standards are required to ensure completeness.
- Acceptance testing planning must be started early in the application system development process.
- The user group must be heavily involved throughout all stages of acceptance testing and must approve the final test results.
- The test team members must maintain a posture independent from implementers.

Most organizations interviewed that used formal acceptance test teams noted that it is expensive to perform thorough testing of major application systems. However, the analysis does indicate that acceptance testing is cost-beneficial over the life of the application systems and there is a definite trend to use formal acceptance testing techniques.

In summary, adequate acceptance testing is very important and it does provide a mechanism for improving the quality of application systems and reducing the number of operation problems that are visible to users and top management. It does not guarantee a 100% error-free system or prevent bad design. It is an important control because it allows users, EDP auditors, designers, implementers, and other concerned parties an opportunity to view the application system in its final form before becoming an operational entity. It ensures processing accuracy, completeness, and reliability.

PROGRAM CHANGE CONTROL

Program change control, a formalized procedure for modifying application and operating systems, requires written approvals and supporting documentation. The objectives of program change controls are to prevent unauthorized and potentially fraudulent changes from being introduced into previously tested and accepted computer programs.

Change Authorization – This includes a formal change request and authorization form and procedure, approval procedures for programmers to gain access to source coding maintained in a source program library, and testing and certification of test results.

Program changes to existing systems are one area in which both internal auditors and data processing personnel have been increasing their efforts at control. Major changes are always preceded by planning of the kind that is associated with major development or acquisition efforts, and the internal auditors' roles are very much like those played in the system development process. Minor changes to existing programs and files pose a different problem because they usually occur with little notice and must be added to "keep the system running." Most organizations have their own special way of handling the problem.

One field interview site has set up a program change committee, through which all modifications to existing systems must pass. This program change committee must also pass on and approve new systems before they gain operational status. This program change committee consists of personnel from programming, operations management, technical service, and the systems and systems programming departments. This committee is meeting daily, but in the future it intends to appoint one full-time program change coordinator who will handle the day-to-day operations of approving program changes and will call committee meetings only when higher authority is required.

The committee evaluates the programmers' testing of the change, evaluates the validity of the change, examines operating instructions, and generally reviews all program changes before final implementation in the production system. This committee has absolute power and no system can be modified without committee approval.

The programmer submits his job control language (JCL), documentation, and changes to the committee for its approval. After being scrutinized and approved by the committee, the JCL, documentation, and changes are passed on to the operations department, which is responsible for modifying the load libraries.

This committee is an active control, but it does not preclude illegal program changes by programmers or systems personnel. To assist in the prevention and detection of illegal changes, the committee has inserted numerous other controls.

One control that is being used is a specialized usage of IBM's Systems Management Facilities (SMF) Reporting System. The SMF was modified to gather data automatically so a report on program changes in the production libraries is printed daily. Another report has been developed that lists all data base accesses that do not come from load libraries. In this way, the committee can have a report that shows potential unauthorized file accesses.

Another control used requires application programmers to submit suggested JCL specifications, and the operations personnel will actually insert all JCL cards in any update or program change. In this way, a programmer cannot embed JCL cards in the middle of a deck on a batch update to illegally modify some other program.

Another control exists for emergency cases in which a programmer must come in after hours to fix a system. The control requires that an emergency change report be sent to the program change committee the next morning. This report must include a comprehensive list of the program coding that was changed, a copy of the console log, and a copy of SYSOUT for the time the programmer was making the emergency changes. (SYSOUT is an IBM term for a class of operating system output that includes the devices that were used for an application run.) Internal auditing reviews the emergency call list to see who was in and when, and checks back to see if the program change committee was apprised of each emergency call.

Management estimates 500 man-hours were required to incorporate the above program change control system. This 500 hours involved up to 100 hours of programming effort to modify SMF and make other minor program changes, and about 400 hours writing the manual procedures and training the people in the use of the manual procedures to effect the program change control procedure.

Block Cut-In – Managing the number of changes can become a burdensome task unless steps are taken to minimize the frequency of change. One large organization used "block cut-in," in which program changes are collected and inserted into application programs as a group so that changes can be traced back to a block if errors occur.

Block cut-in of changes improves application control and control over operating systems. Unauthorized program changes can be detected more easily, whether they are operating-system or application-program changes. Finally, block cut-in improves the efficiency of general operations by batching changes and installing them into the system as a unit.

To implement block cut-in, data processing operations must do the following:

- Obtain agreement from the users that systems will be updated only a limited number of times a year.
- Develop a version numbering scheme to denote the releases (e.g., Payroll-Release 23).
- Develop a means of collecting and holding program changes as they are tested and approved, but before they are implemented into the system.
- Schedule cut-in dates and announce them, so a user has the option to change priorities of enhancements to be programmed and tested.
- Cut-in collected program changes at one time and announce the new release of the system.

This approach has two limitations. First, since only one to six releases per year are planned, some program changes might normally be held for several months. For program changes that are required quickly, such as a payroll withholding tax rate change, this may be impossible. The alternative is to have a variable cut-in schedule or a one-time emergency cut-in procedure. Second, when one change is made on a certain date, it may necessitate making changes in other related systems. This may mean departing from the planned schedule.

Program Packages – Some organizations interviewed used program packages to control changes to systems libraries. The packages allow for password protection of source libraries from which load libraries are created.

In summary, most organizations interviewed felt they could and would improve their program change controls and the procedures for using the controls. Program change controls allow data processing personnel to maintain the integrity of accepted systems and develop change lists that allow for traceability of the evolving application systems. Change controls frustrate programmers and sometimes cause delays in fixes to application systems. Program change controls are beneficial because they encourage data processing personnel to exercise more caution as they make changes to accepted production systems. SRI found a trend toward the use of some type of formal program change control procedures, but most organizations appear to be slow in moving to extensive use of this procedure.

DOCUMENTATION

Documentation is the process of describing on paper what functions an application system performs, how it performs them, and how the functions are to be used. The objectives of good documentation are to provide application system designers, implementers, testers, users, and EDP auditors with a clear means of understanding all aspects of the application system.

Documentation is important because it provides the primary communication channels between users, EDP auditors, and data processing personnel. It is important because it helps ensure correct and efficient processing within both data processing and user areas; it increases the ease and accuracy of computer program maintenance, and it provides internal auditors with an independent basis for evaluating application control.

System documentation occurs throughout system development. In the early stages, it is used to document system objectives and definitions. As such, it can be used within the system project staff and can serve as a vehicle for approval from units of the organization outside the data processing department. In the system development control approach, using control points discussed earlier in this chapter, we see this form of documentation used for approvals throughout the process.

Control elements provided by a comprehensive documentation program include the following:

- System documentation, including a clear statement of the system objectives, a flowchart showing the flow of information through the system and the interrelationship between manual and computer processing steps, system specifications that governed design and development, input forms and procedures, record formats and descriptions, descriptions of audit trails, and balancing and control procedures.
- Program documentation, including descriptions and flowcharts necessary to allow efficient and accurate maintenance subsequent to initial installation. Such documentation normally includes control cards and JCL, program listings, program test data, a testing log, input/output distribution instructions, data retention instructions, console operator instructions, and copies of program change request forms.
- Operations documentation, including instructions that provide the computer operations function with adequate instructions to accurately and efficiently run a computer application, balance inputs and outputs, and distribute reports. Restart and recovery procedures are also included. The requirements for operations instructions vary greatly from installation to installation depending upon the operating systems and the library procedures, and input/output controls that have been adopted.
- Library documentation, including procedures for backup, retention, restrictions on access to sensitive data, and inventory record keeping.
- User documentation, including a narrative description of the system's operation accompanied by a general flow diagram.

This basic user documentation is supplemented by specific instructions governing the proper completion of input forms and transactions. Control procedures provide users with instructions they need to balance, reconcile, and maintain overall control transactions, masterfiles, and the results of computer processing.

One approach to documentation control is used by a large food manufacturer. This organization makes use of a documentation test to be administered throughout the phases of system development. The primary purpose of the test is to ensure that appropriate documentation exists for the following major phases of system development: feasibility study, system proposal, system specifications, and installation, including turnovers to data processing and to program support.

In addition to verifying the existence of sufficient documentation, this technique also assesses various aspects of the documentation, including the following:

- Does the documentation give evidence that processing controls will be adequate?
- Have sufficient controls been built into the system to allow effective operation and maintenance?
- Can the documentation be used as a basis to prove that controls over operation and maintenance are adequate?

Within each of the major phases of system development, a detailed list of minimum documentation requirements is provided. For each phase, the required set of documentation must be present and assessed before the next phase can begin. The documentation test embodies a variety of levels of documentation to be used to control the actual overall processes and detailed control documents to be used within a phase. As such, it serves as an inventory for the documentation that should be present following actual system development and installation.

General documentation trends and concepts discovered during the study include:

- A desire by data processing personnel to develop application systems that are more self-documenting through use of techniques such as structured programming.
- A realization that documentation must be controlled. One company had gone so far as to require limited copies of sensitive documents and to require signatures each time a person is given a copy.

In summary, application system documentation is an important control technique because it provides a mechanism by which persons not directly associated with the development process can understand and evaluate the application system for usage and controls. Particularly important to internal auditors, it allows them to audit applications with general independence from data processing.

DATA BASE ADMINISTRATION

A data base is a collection of interrelated data stored together with controlled redundancy to serve one or many users or applications in an optimal fashion. In addition, the data are stored so that they are independent of programs that use them. The data base administrator (DBA) is the manager responsible for coordinating and controlling the design of the data base management system (DBMS) as well as the user interface and access to the data stored within the DBMS.

Data base administration and the role of the DBA are discussed in this chapter because the field interviews clearly indicated a trend toward the use of data base administration techniques to control data access and the early stages of the application system development process.

The objectives in creating a data base are to reduce the amount of data and the programs required to meet corporate needs, to control the user interfaces and access to the data, and to manipulate the data in the most efficient manner possible.

The objective of data base administration is to provide a focal point for defining and controlling all elements of data base systems. This requires that the DBA take an active part during the system development process, especially the early stages of it, to ensure that adequate controls exist.

The particular functions performed by a DBA have existed in one form or another for many years. Applications-oriented systems have been developed with the data base defined in direct support of the procedures. This is opposite from the data base design approach, in which the center of attention is the data. As systems become increasingly integrated, the importance of a data base design approach becomes critical to the successful design and development of integrated information systems. A data base environment supports a data base design approach. It is thus the management of the data base environment that will determine the success of a data base design approach and the resulting integrated corporate data base.

The Primary U.S. Mail Survey showed that, of the organizations using DBA techniques, over 70% used the DBA function at varying stages throughout the application system development process.

The five major areas of responsibility of the DBA as determined during interviews are data definition and data base design, administration, operations, system monitoring and improvement, and general support.

Data Definition and Data Base Design – The DBA is responsible for all standards related to the data base. These standards cover how data are defined and ultimately how data are stored, protected, and accessed by users of the data base.

Administration – The DBA enforces management's policies for disposition of the organization's data. The specific areas involved in this activity include the administration of data base standards used for data definition, use, protection, access, and documentation.

Operations – The DBA ensures that all resources within the data base environment operate in the most effective manner as the system becomes operational. The DBA is responsible for working with computer operations in developing both formal and documented procedures for operating the data base. These procedures include overall scheduling of computer time to ensure priority use of the data base, utilization of data base components, and general operational use of the data base. The DBA is also responsible for seeing that the system restart and recovery procedures are properly exercised and controlled.

System Monitoring and Improvement — The DBA continually monitors and measures the various aspects of performance to ensure an efficient level of service while maintaining data base integrity.

General Support — The DBA is the primary interface between the organization and computer vendors supplying hardware and software in support of DBMS. He must assess new products and their relative value to the data base environment.

The DBA conducts the necessary internal education. Education includes programs for new users and present users to ensure that all components and standards of the data base environment are being effectively and properly used.

In summary, both data base and data base administration are receiving increased attention in many large institutions. Data base use is growing because organizations are placing more types of related data into computer systems and the use of data base management systems provides the best means of control. Data base administration is growing to meet the requirement for controlling data bases and data base management systems. Data base systems place heavy emphasis on the data to be processed. Data base administration provides a centralized mechanism for controlling the definition of data and the access to and protection of the data, as well as a means for controlling the development of the application system. The concept of data base administration is relatively new, and consequently there is not a universally accepted method for its use. However, data base administration is a good control because it does encourage organizations to view application system development from the point of view of how the data are to be used.

SUMMARY

[illegible] system [illegible] important because the ade- [illegible] effectiveness of controls included in computer application systems are heavily affected by the [illegible] procedures used during the system [illegible] process to ensure the accuracy, completeness, and validity of data being processed by complex computer application systems, internal auditors must consider an entire system of controls that encompass data preparation, data entry, data communications, data processing, and data distribution. To accomplish this goal, internal auditors are being required to participate actively in the system development process because retrofitting controls, or trying to understand internal controls after the application system has been implemented, is costly and difficult. In addition, internal auditors are responsible for evaluating and verifying operational and user controls associated with the computer application systems. Participating in the system development process broadens the internal auditor's perspective and understanding of the computer application systems and allows him to perform all his functions more completely.

The primary controls and techniques of concern during the system development process are documentation, project management, and application system change controls. Documentation is one of the most important controls from the internal auditor's point of view because it provides him with a means for conducting independent reviews of system specifications and design, system testing and operation requirements, and user interface and usage specifications. Matching all areas in an independent but coordinated manner allows for completeness. The project management technique provides for a management structure that forces periodic status reporting to keep all parties informed. These status reports allow the user and internal auditors to monitor progress and costs against plans and budget. Change controls are important because most computer application systems are large and complex and require periodic changes. After an application system has undergone formal acceptance testing, the integrity of the system must be maintained as it is changed.

The systems development life cycle technique is a rapidly growing trend in data processing because it encompasses the primary concerns of data processing and encourages the development of better application systems. System development controls and techniques tend to be more expensive than many other controls because of the requirements for skilled persons and new concepts. For example, SDLC is a technique that requires EDP auditors, users, and project management to meet on a regular basis, communicate technical ideas, and receive formal approvals from each other. The cost of system development controls must be weighed against the cost of the computer application system over its expected life. While the controls or techniques may be expensive over a short time, the cost over the expected life of the system is much more reasonable. Application systems developed with good management controls and techniques will require less maintenance and be less susceptible to fraud or other abuses.

GAO - IMTEC
INFORMATION CENTER

COMPUTER SCIENCE & TECHNOLOGY:

Testing Background Information
- Phases of lifecycle involving testing - Pg 28
- Guidelines for testing - Pg 33
- - Plan for a shakedown period - Pg 37.

COMPUTER SOFTWARE MANAGEMENT: A PRIMER FOR PROJECT MANAGEMENT AND QUALITY CONTROL

IMTEC INFORMATION CENTER
LIBRARY
MASTER COPY

NBS Special Publication 500-11
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

PREFACE

The National Bureau of Standards is the primary technical advisor on computers for government agencies (Brooks Act, Public Law 89-306). NBS assists Federal agencies in specific projects, and so obtains first-hand experience with agency problems and competencies. This guide was prepared drawing upon extensive contacts with government computing personnel, and using the published recommendations of authorities in software management. A variety of executive guides and primers in other fields were reviewed in choosing the scope and approach to the material covered here. Other staff members of the Institute for Computer Sciences and Technology assisted by reviewing the material as it was developed. The author was helped particularly by the constructive comments of I. Trotter Hardy and Theodore Linden.

NOTICE TO READERS

This guide is intended to become a standard reference for managing Federal government software projects. Comments by the readers on its completeness, clarity, etc. would be valuable in making appropriate revisions for this purpose. A postage-free reader evaluation form is included at the back of the report, and you are encouraged to complete and return it.

COMPUTER SOFTWARE MANAGEMENT:
A primer for project management and quality control

Dennis W. Fife

Today, providing computer software involves greater cost and risk than providing computer equipment, because hardware is mass produced by industry using proven technology, while software is still produced mostly by the craft of individual computer programmers. This brief guide is intended for managers who are responsible for computer projects, to explain the use of quality controls and software management methods. The typical problems of software development are explained. Over twenty distinct quality controls are defined, and recommendations are given for software management actions. Empirical information is included that would help top executives to appreciate the potential problems and importance of software management.

Keywords: computer management; computer programming; computer project control; computer software; software engineering; software quality; software reliability.

WHY THIS GUIDE WAS WRITTEN

Computer software is the computer instructions or programs, as well as the data files and descriptive manuals, that have to be provided to make computer equipment (hardware) perform useful services. Today, providing software involves greater cost and risk than providing the equipment, because hardware is mass produced by industry using proven technology, while software is still produced mostly by the craft of individual computer programmers and users.

Software management is the technical and management control throughout the life cycle of all software that supports an organization's mission and objectives. Software management is effective when high quality software is being obtained with reasonable and controllable costs. Software management is inadequate when poor decisions and lax technical controls result in faulty software performance, computer service fiascos, or project cost overruns. The unabated high cost of software as well as many publicized cases of software catastrophes are convincing evidence that software management is poorly understood and practiced in most organizations.

The problem is deep-seated and serious, because many managers who control software projects do not have professional computing expertise, and many programmers lack the technical skills to properly design and produce the highly complex computer software needed today. Equally important, the expansion of computer applications over the last twenty years has far outpaced the evolution of basic principles and engineering discipline in software design and production. Much more applied research is needed before software engineering will progress to the effectiveness found in other engineering and production fields.

WHO THIS GUIDE IS FOR

This guide is intended for managers and staff who are responsible for computer projects. These will be data processing professionals and supervisors primarily. But the intended audience also includes executives and managers who were not trained as computer professionals and who perhaps have never used a computer themselves. Every reader will need to understand the rudiments of computer operation and programming.

WHAT THIS GUIDE IS FOR

The objective of this guide is to give the reader an appreciation of the software pitfalls in computer projects, and to explain the use of quality controls and software management methods. The typical problems of software production are highlighted, and recommendations are given for setting up software management procedures.

WHAT THIS GUIDE IS NOT FOR

The brevity and special purpose of this guide mean that it is unsuitable for other important purposes. In particular, this guide is NOT a survey of programming methods, NOR a handbook for the direct supervision of computer programmers. Its use cannot compensate for a lack of technical expertise that is essential to direct management of software design, development, and maintenance. Finally, this guide is NOT a popular treatment to arouse the interest and concern of the general public. It is expected that readers have a professional interest and obligation in decision-making for software projects.

HOW TO USE THIS GUIDE

The sections marked by (#) in the Table of Contents contain specific recommendations and guidelines for effective software management. This material should be utilized to develop appropriate administrative orders or management procedures to implement the recommended practices in all software projects. Selected references for further study of the more important topics are given at the end of this guide. Additional sections of this guide contain information that would benefit a senior executive in understanding the problems and importance of software management.

WHY SOFTWARE MANAGEMENT IS IMPORTANT

*Software is a very expensive resource for government.

The procurement and development of computer software is "big business" for many organizations, including the government. Estimates of Federal government spending range as high as as $5 billion annually on software, and the accumulated investment in current Federal software probably exceeds $25 billion dollars. Software costs greatly exceed equipment costs over the useful life of computer services, and the initial development costs for government software systems are growing to enormous levels. For instance, cost estimates for the newly proposed Tax Administration System of the Internal Revenue Service are well over $500 million dollars, and the Air Force has estimated its software costs for command and control systems in the 1980's will be several billion dollars.

*The government software inventory is large and complex.

Among the more than 8,000 government computer installations, many have 500 or more programs for specific government tasks, in addition to others acquired from the computer manufacturer for general support of computer operation. Government computer applications, which include sophisticated military systems as well as public service systems handling millions of accounts, are typical of the most challenging uses of computer technology within the state-of-the-art.

*Software flaws and mismanagement can be ruinous to government projects and public services, and even to national goals.

Government and business are completely dependent on computer services today. Manual methods are no longer considered because human means cannot be organized to handle the volume of data and short deadlines that are needed in widespread public services. Social and economic activity is geared to the performance of computer systems. Examples are many, as in airline reservations systems or the Social Security Administration systems that distribute over $50 billion annually to about 25 million individuals. These benefits are possible only with effective management of the computer software. In recent years, recurring cases of major software failures have demonstrated the serious consequences of software mismanagement to legislators, journalists, and the public. One dramatic case occurred in 1975, when a government agency made more than $500 million in overpayments to individuals as a result of erroneous computer programs.

*No technology or standard practice now exists that will surely prevent faulty design, logical errors, cost overrun, or late delivery for any software project.

The preparation and maintenance of computer software is a laborious, human task that can require the coordinated effort of dozens or even hundreds of programmers. The possibilities for program errors, poor design, or poor decision-making may be higher than in any other technical field, because of extreme technical complexity and the lack of standards and common practices in this new technology. The speed and volume of data handling by the computer creates disastrous consequences from errors in programs and accentuates the difficulties caused by only minor design flaws. There is no practical method known that will guarantee software as free from errors or design flaws. Although programming productivity has been improving, the advances have not kept pace with the tremendous growth in computer applications and the dramatic gains in performance versus cost for computer hardware over the past twenty years. There are very few standard computer programs for common applications. There are no standards covering common work practices and technical goals, such as the measurement of software reliability. Thus, every software project is a unique effort. Success or failure is determined more by the individual technical skills and management capability of the personnel involved, than by any other factor.

WHY EVERY AGENCY NEEDS SOFTWARE MANAGEMENT

*Software management must be the responsibility of the computer user organization.

Over 70% of all software is developed by the users of computers, rather than the manufacturers of computer equipment or the producers of commercial software. Although the computer industry is working to improve software management for its products, industry cannot be expected to provide a comprehensive solution, nor to cover the unique circumstances of each computer user.

*Software design and programming is a complex technical activity that must be directed effectively.

Although many people nowadays may understand the rudiments of programming from college or even high school courses, the simple homework exercises in such courses do not represent the "real world" of professional practice. The design and production of numerous programs that must work together for a major service, such as the payroll of a large business, is not routine effort that can go unsupervised or be done by junior personnel. The funds and personnel resources involved may be a major investment for the organization, and the technical choices may drastically affect non-computer work processes in management, clerical, and client relations areas. Technically complex programs may demand advanced professional skill which is very scarce and must be carefully directed at the critical technical problems. Management and technical control by a professional software designer is essential for resolving complex design issues and for directing programmers to produce a specified product within cost and schedule targets.

*Software management establishes accountability for project decisions and objectives, and provides upper management with visible measures of progress toward desired goals.

Software projects usually are initiated by managers in such areas as payroll, engineering, plant and supply, etc. Although they may control the project budget and schedule, and have the best information on potential benefits and requirements, they are often unprepared to conduct a design project or to judge the results provided by an outside technical group. Thus, software designers sometimes work independently using their own judgement, with little pressure to reliably estimate their progress and the remaining work or costs. Software management includes checkpoints to assure that important objectives and issues are carefully evaluated by the concerned parties, especially in regard to cost,quality, and schedule, and that users and designers take their appropriate responsibility for the decisions made.

*Software management provides the technical controls and resource management to produce high quality software within acceptable funding and time limits.

Software management emphasizes technical criteria and working procedures to resolve the technical issues that are critical to the success or failure of a project. Major questions commonly arise concerning the feasibility or performance of a desired system, the attainable quality of the system considering the resources available to produce it, and the methodology for design and testing in order to achieve the most effective software product. Software management delineates quality controls, standards, planning factors, and experience data which help designers and programmers to organize and direct their efforts efficiently in solving such problems within available cost and time limits.

*Software management controls costs after software operation begins.

The useful life of software for a significant service typically is over eight years. Experience has shown that continual changes are necessary in software, to correct previously unknown errors, to improve useability or performance, or to include new functions required because of legal, administrative, or technical changes. As much as 70% of software cost occurs in this redesign and maintenance, after software has been delivered and put into use. It is important to recognize that maintenance is no less challenging than development, for change to programs involves new design work and should be done according to planned cost and time goals. Software management procedures, consistently applied throughout the software life, will make best use of limited funds by stressing close management of effort toward the most needed software capabilities.

BASIC TASKS OF SOFTWARE MANAGEMENT

*Software management consists of all the technical and management activities, decisions, and controls that are directly required to purchase, produce, or maintain software throughout the useful life of a computer system or service.

Software management is primarily concerned with computer programs, data files, and documentation, rather than the computer equipment configuration that executes the programs. Even so, software designers must have a say in hardware decisions if they are to meet the goals for a computer service. The term "system" will be used here for the combination of hardware and software that provides a computer service. Since software has little purpose without hardware, the terms "system" and "software" may be used interchangeably without causing confusion.

The basic activities and decisions are best introduced in relation to the commonly recognized phases of a system's life cycle, shown in Figure 1. The INITIATION phase involves the assessment of an existing, inadequate system, in order to determine the feasibility of replacing it with an improved system. The four phases, DEFINITION through TESTING, are the development period, where a new or improved software product is being designed and implemented. The OPERATION phase starts with the operation of a new system by its intended users, and includes the subsequent maintenance and minor redesigns that user experience may dictate. Eventually, maintenance no longer suffices for needed improvements, and another development cycle is begun.

Specific guidelines for each phase will be given after the following summary of the tasks included. Figure 2 helps to understand each phase by summarizing the documents produced that are needed for quality control.

*In the INITIATION phase, software management assures that the technical designers understand the purpose and scope of the envisioned system, and adequately perceive its required functions.

The INITIATION phase begins when management recognizes that a need exists for a new or improved computer service, and that a study should be made to formulate a software solution. During this phase, software planners or system analysts identify the intended users and investigate their work processes. The planners describe the services needed from the envisioned software, and then conceive and outline alternative software solutions. A close working relationship with the intended users is essential. The alternative approaches should be described in basic terms that users can understand. User concurrence with the stated requirements and the recommended software approach is mandatory.

*The INITIATION phase assures that the recommended software approach is technically and economically feasible.

An estimate must be made of the costs and benefits for each of the potential software solutions. This will lead to a recommended solution and provide data regarding feasibility. Comparable existing systems are investigated to derive cost data and to confirm that proposed solutions are technically realizable. Any pitfalls, failures, or questionable design areas in comparable projects should be covered thoroughly. Comparisons of the proposed project to past failures or problem areas would assure that they can be avoided or resolved. General specifications must be written to show clearly the scope and character of the recommended solution. The INITIATION phase concludes with a report delineating the recommended software and the supporting analysis. Management may then approve the concept and budget the recommended funds and personnel for acquisition and operation of the planned system.

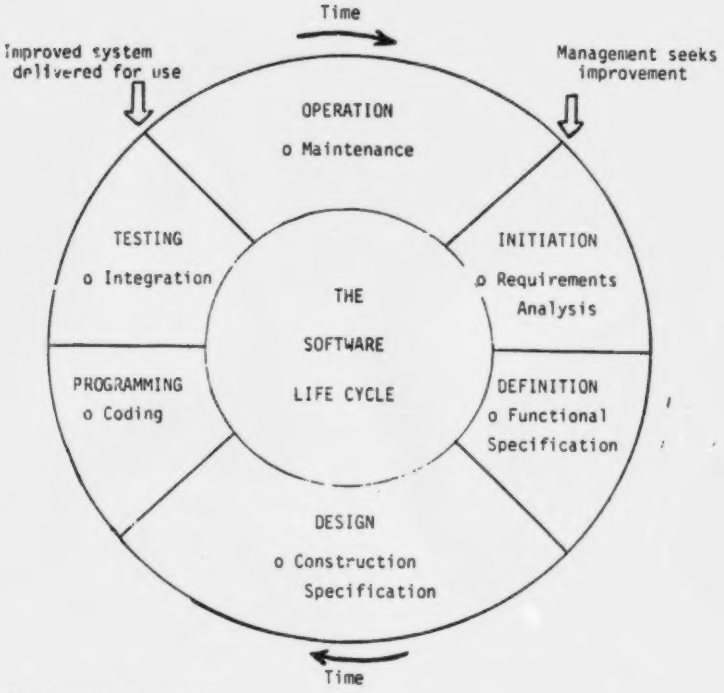
*The DEFINITION phase defines the functional and performance requirements, in order to confirm that the software, when built, would meet its objectives.

The DEFINITION phase begins when management commits resources to the design or purchase of the proposed software. In this phase, detailed specifications are created for those externally apparent functions and design characteristics that are crucial to the purpose and operation of the software in the intended environment. Input and output data, processing operations, and performance goals are defined. The Functional Requirements define what the software must do and the general aspects of how it will be constructed, rather than the details. The DEFINITION phase concludes with a specification that is sufficiently precise for outside contracting, if desired. Equally important, the specification allows the intended users to confirm that the proposed system will meet the need and will have acceptable qualities of useability, generality, etc.

*The DEFINITION phase provides a project plan for managing the acquisition and installation of the system.

A thorough plan for producing or purchasing the software is prepared as part of the DEFINITION phase. Projections are made of the detailed costs. A delivery schedule is laid out, including intermediate milestones and technical reviews for evaluating progress and insuring that the project will meet its goals. Working methods are defined for quality control of intermediate and final products. The plan should extend to the installation and initial operation of the system, and should include user training, document preparation and review, acceptance testing, and the continuing obligations of any contractors, vendors, or support organizations. The Project Plan is extremely important because it specifies how the effort will be managed, how problems will be resolved, and how the quality of the final system will be assured.

Figure 1. The Life Cycle of a Software Product


Time
Improved system delivered for use
Management seeks improvement
OPERATION
o Maintenance
INITIATION
o Requirements Analysis
DEFINITION
o Functional Specification
DESIGN
o Construction Specification
PROGRAMMING
o Coding
TESTING
o Integration
THE SOFTWARE LIFE CYCLE
Time

o Major technical activity

Figure 2. Chart of Basic Quality Control Documents

Life Cycle Phase	Documents	General Content
Initiation	Feasibility Study Report	Requirements analysis, definition and evaluation of alternative solutions, recommended software concept.
Definition	Functional Requirements	External specification of software functions and operation, including design constraints.
	Quality and Performance Requirements	Desired design attributes and quantitative performance parameters for key processing operations.
	Project Plan	Cost and work breakdown, detailed schedule including quality assurance milestones, and definition of primary management and quality control methods.
Design	Design/Coding Standards	Detailed design attributes and coding conventions for quality.
	Design Specifications	Software architecture, module definitions, interface specifications, and programming requirements.
	Test Plan	Testing criteria, testing schedule and work breakdown, standard test case specifications.
Programming	Review Reports	Identified discrepancies and recommendations from team reviews.
	Unit Test Reports	Results of unit test for each module.
	Inspection Reports	Results of inspection sessions for each subsystem
	Software Manuals	Usage, operation and maintenance description:
	Program Listings	Source language statements for each tested module
Testing	Test/Fault Report	Results of scheduled tests.
Operation	Fault Report	Symptoms, circumstances of detected or suspected failure.
	Specification	Functional and design specifications for maintenance and improvement work.
	Maintenance Plan	Work breakdown, cost estimate, and schedule for proposed rework.

*The DESIGN phase produces a thorough design specification for guiding a timely and problem-free implementation of the proposed software.

The DESIGN phase begins when the functional requirements have been validated by intended users, and implementation of the system is approved. The DESIGN phase results in detailed specifications of the internal construction of the software, for use by programmers who will implement the design. The DESIGN phase involves definition of the internal architecture of the software, performance trade-off analyses of algorithms, specification of individual programs or modules, specification of interactions between components of software, etc. The Design Specifications are important as a means for resource management and quality control during PROGRAMMING.

*In the PROGRAMMING phase, software management assures that appropriate technical choices are made, and that high quality workmanship is applied in producing code meeting the Design Specifications.

The PROGRAMMING phase often is merged with DESIGN and so may have no distinguishable starting point. PROGRAMMING involves final technical choices for internal program design and the creation of programming language statements or "code" that is executable on the computer and that implements the design. Software management here is the day-to-day supervision of programmers, and guiding and reviewing their technical results. PROGRAMMING effort includes the testing by each programmer of his individual programs and the preparation of pertinent management reports and software documents. Test reports, inspection reports, and program documentation act as tangible quality controls over PROGRAMMING accomplishments.

*In the TESTING phase, software management assures that no significant errors or design discrepancies will impede usage of the system in the intended operational environment.

The TESTING phase is concerned with verifying that the software product meets the functional specifications and contains no significant discrepancies or logical errors to impede its use. TESTING proceeds according to a Test Plan prepared in the DESIGN phase, and focuses on the integration of individual programmer results in an overall working system. Noted discrepancies and errors are documented for corrective action by the programming team. Gross shortcomings in the original design or functional specifications may be exposed; this should lead to a major project review in order to plan and organize additional definition and design work.

*In OPERATION of a delivered system, software management assures that maintenance is as well managed and controlled as original development.

The OPERATION phase follows the delivery and installation of the software in the field, and involves periodic redesign and improvement. Software management procedures that are recommended during development can be streamlined to provide similar close control of the reduced effort committed to corrective design work. Thorough specifications, quality workmanship, effective testing, and management review remain highly important in directing technical effort to meet cost and schedule targets.

*During the development period, the earlier activities may be continued or repeated in order to remedy design defects and improve quality of the final system.

Although the successive phases defined here should be followed whenever practical, some flexibility is advisable, especially when a project involves unusual complexity or innovation. For example, it may be important to carry only a part of the system through to DESIGN and PROGRAMMING, in order to confirm feasibility before beginning DEFINITION for the entire system. Or, with limited effort available, development of some supporting capabilities or auxiliary programs may be deferred until the mainstream processing is successfully completed. So, all parts of the software need not be at the same stage of development, and the life cycle orientation may be viewed with some flexibility.

THE QUALITIES OF SUPERIOR SOFTWARE

These general properties are commonly accepted as characteristic of high quality software. The techniques of software management may be used to control quality at acceptably high levels or to maximize quality insofar as practical with the resources and time available for a project.

CORRECTNESS--Programs perform exactly and correctly all the functions expected from the specifications, if available, or else from the documentation. This means that incorrect documentation is as serious as an incorrect program. Correctness is an ideal quality, that is rarely determinable, so a more practical quality is RELIABILITY.

RELIABILITY--Programs perform without significant detectable errors all the functions expected from the specifications or the documentation. High reliability indicates that programs are relatively trouble free in performing what they are claimed to do. But an equally important question is whether the functions and performance are adequate and suitable to a needed purpose. The latter quality is called VALIDITY.

VALIDITY--Programs provide the performance, all functions, and appropriate interfaces to other software components, that are sufficient for beneficial application in the intended user environment. This means the software, without additional programming or manual intervention, has the capabilities that reasonably would be expected for its purpose. Validity is a quality of specifications as well as computer programs. Examples of an invalid program would be an interactive editor that had no online function for retrieving stored text for inspection, or a FORTRAN language compiler that had no DO loop implementation. Validity involves judgement of user requirements, and may change if the intended application or purpose is altered. Because poor reliability may render a needed function useless, reliability is necessary to validity.

RESILIENCE (or ROBUSTNESS)--Programs continue to perform in a reasonable way despite violations of the assumed input and usage conventions. Input of unacceptable data, or an inconsistent command, should never cause a result that is astonishing and detrimental to the user--such as the deletion of any valid results obtained previously. Programs should include routine checks and recovery possibilities that are "forgiving" of common user and data errors. Resilience is related to the broader quality of USABILITY.

USABILITY--Programs have functions and usage techniques that are natural and convenient for people, showing good consideration of human factors and limitations. For example, the programs have few arbitrary codes for data in input or output, have consistent conventions in different operating modes, and provide thorough diagnostic messages for errors or violations of use.

CLARITY--The functions and operation of the programs are easily understood from the user manual, and the program design and structure are readily apparent from the listing of program statements. This means that documentation must be well written, but also that the program is carefully designed, with meaningful choices of variable names, use of known algorithms, frequent and effective comments in the program to describe its operation, and a modular structure that isolates separate functions for examination.

MAINTAINABILITY--Programs are well documented by manuals and internal comments, and so well structured that another programmer could easily repair defects or make minor improvements. Clarity is essential for maintainability. But also implied are a wide variety of good design attributes, such as program functions that help to diagnose potential problems, e. g., periodic reports of status or control totals; or, general techniques that can be readily adapted for change, e. g., the isolation of constants, report titles, and other static data as named variables.

MODIFIABILITY--Program functions that might require major change are well documented and isolated in distinct modules. Maintainability is essential to modifiability. But modifiability means that a concerted effort was made to anticipate major changes, and to plan the software design so that they could be made easily.

GENERALITY--Programs perform their functions over a wide range of input values and usage modes. Programs are not limited to special cases or ranges of values, when the functions are commonly or reasonably extendable to a more general case.

PORTABILITY--Programs are easily installed on another computer or under another operating system program. Standard programming language is used, and hardware or other software dependent features are isolated for easy change.

TESTABILITY--Programs are simply structured and use general algorithms, to facilitate step-by-step testing of all capabilities.

EFFICIENCY/ECONOMY--Programs have high performance algorithms and conservatively use computer resources, such as main storage, so that the cost of program operation is low.

ORGANIZING FOR SOFTWARE MANAGEMENT

*A project team is the basic organization for software management.

The team approach is effective because a mix of needed skills can be quickly assembled from an organization's available talent, and the most talented designers can be consistently utilized on the toughest problems. Team practices provide better oversight and counseling through peer reviews, better dissemination of technical and status information about software modules, and better back-up of personnel. The project team approach will be an unusual practice where computer personnel have been organized according to specialties such as "operations", "system analysis" and "systems programming". Nevertheless, the advantages gained by forming an appropriately skilled team, dedicated to one product objective, warrant a departure from the traditional organization. A project team may be formed temporarily to carry out software development, with further effort after installation handled in the normal organizational framework. However, the advantages of team work apply to maintenance effort in the OPERATION phase as well as to the development phases. So a project team may be advisable when major maintenance or improvement activities are done.

*For projects involving up to about 10 personnel, one individual should be responsible for the entire system design and the project management.

This concept has been called the "Chief Programmer" approach, for the individual must be an exceptional programmer as well as a very competent manager. Control through one individual, especially from DEFINITION through TESTING, offers a better possibility of a consistent, high quality product, than would cooperative work by several individuals or smaller teams. The team chief personally defines and designs the entire system, and produces many of the critical modules. The chief exercises complete authority over the technical results and team production. He or she represents the project with users or customers and with higher management, and decides the project commitments made to them. The chief controls personnel assignments and schedules, and has supervisory responsibility for all team members. The team chief usually should have a minimum of five years of diversified, intensive experience in system analysis and software design, and should have demonstrated the personal qualities, such as judgement and maturity, that are needed for effectively supervising other specialists and for dealing with clients and superior officials.

*The project team should include an assistant chief, specialists for key problem areas, and supporting personnel.

The assistant team chief substitutes for the chief enever necessary, and serves as his primary technical dvisor. The assistant may personally define, design, and rogram some parts of the system. When needed, particular pecialists become team members to handle difficult design roblems requiring their unusual knowledge and experience. n PROGRAMMING especially, a programming language specialist hould be available to answer questions about the effect and erformance of various instruction sequences, and to suggest etter ways of programming certain algorithms. Specialists n program production tools, testing, and data management echniques are advisable also. The team secretary assists n clerical tasks aimed at improving the team productivity, uch as the preparation of computer runs, purchase of omputer supplies, handling of project records and software ocuments. Additional support is advisable also in ocumentation and technical writing, and in administrative upport of the team on personnel matters, training, anagement information support, etc. Figure 3 depicts a epresentative team structure.

Team techniques are appropriate also for very large projects.

When more than about ten people are needed to do a project, one individual cannot effectively supervise all of them, so the design and programming must be distributed among several teams. Nevertheless, one manager should be given authority over the entire effort, and should make the fundamental decisions on work breakdown, resource allocations, and software design. This project manager and his subordinate programming managers could be viewed as comprising a team to carry out system design and project management. Their working procedures should be similar to those within a subordinate programming team; for instance, conducting periodic design conferences where each team chief presents the designs and other results of his group. The project manager should exercise considerable technical and design leadership, and major aspects of overall system design would be decided by his team. Support arrangements similar to a programming team are advisable, as suggested in Figure 4.

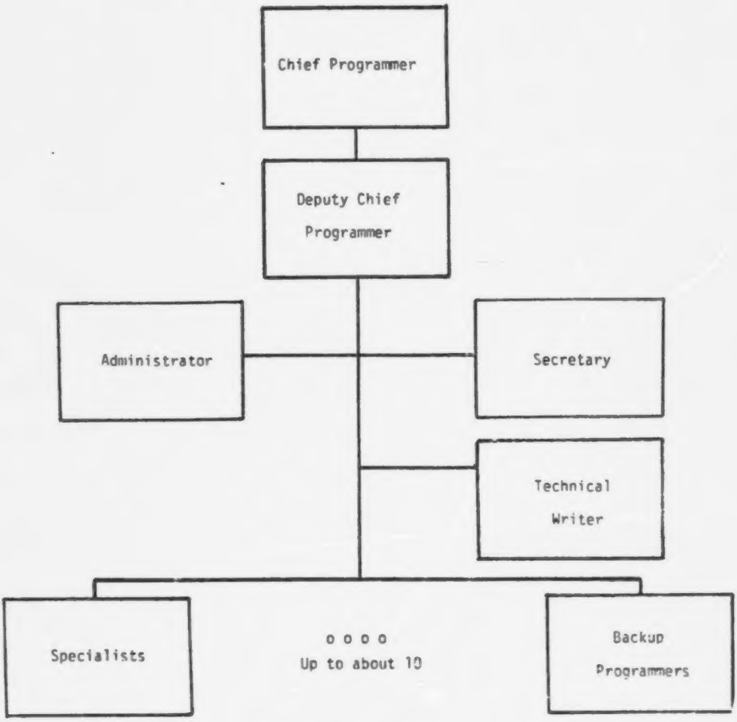
*Team techniques are applicable in managing contracted developments.

A team approach is advisable also when the software development will be accomplished principally by private contractors. The project manager or team chief will serve then as the contracting technical monitor. The contracting team, rather than designing and producing programs, will prepare the technical statements of work, and other contractual requirements such as the quality and performance specifications, and the project plan in regard to mandatory schedules, deliverable products, and quality controls. Technical specialists in particular areas such as testing and performance evaluation are also necessary on the contracting team, and special administrative support is also advisable to improve the quality and timely completion of contract requirements, technical reviews, acceptance testing, etc., as the contractors proceeds with the development.

*Agency-wide direction and evaluation of software policy and quality control is needed in addition to individual team management.

An upper level of software management authority is advisable to provide consistent direction and support for individual teams with regard to organizational policies, standards, and objectives in software. Such an individual or office can contribute significantly in assessing software technology for project needs, and in reviewing project accomplishments with a view to standardizing team procedures, technical aids and software production tools, and even specific software components that have wide utility for the organization.

Figure 3. A Representative Team Organization


Chief Programmer
Deputy Chief Programmer
Administrator
Secretary
Technical Writer
Specialists
o o o o
Up to about 10
Backup Programmers

o Language expert
o Testing specialist
o Tools specialist

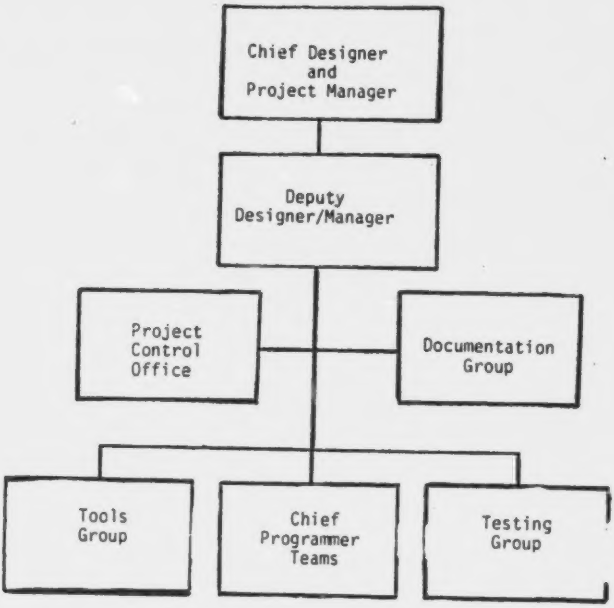


Figure 4. A Representative Organization for a Large Project

Chief Designer and Project Manager
Deputy Designer/Manager
Project Control Office
Documentation Group
Tools Group
Chief Programmer Teams
Testing Group

[redacted]

*Software problems originate most frequently in the analysis of requirements and the design of the computer software.

Nearly 50% of software flaws and errors exposed in final testing or initial use of a new system are traceable to shortcomings and mistakes in the early stages of a project, i. e. from INITIATION through DESIGN. Most authorities suggest that added effort in the early stages is the best way to improve the quality of software. The effort up to the start of PROGRAMMING should comprise about 40% of the total development effort. A greater allocation is advisable, provided working methods and review techniques are geared to strong improvements in the validity and detail of the Design Specifications.

[redacted]

Quality control in these early stages is necessarily judgemental, but will be aided considerably by standardized documentation practice. The National Bureau of Standards has prepared guidelines, FIPS PUB 38, that indicate the content for functional requirements, project plans, and other software documents. Additional guidelines and standards will be issued in the near future. The use of computer-aided text editing and document formatting will be especially helpful in rapidly accommodating numerous small changes in specifications during repeated and lengthy reviews.

*Start a Project Record with the INITIATION phase and continue it over the system life.

The Project Record should be an open record of the major decisions and of the pertinent documents and specifications. It should be official, to clearly indicate firm commitments, yet openly available, so that inadequacies can be detected as early as possible and so that current goals and constraints are apparent to all. This record is distinct from a technical notebook, which also is often effective as an informal medium for exchanging tentative proposals among designers and programmers.

*In the INITIATION phase, identify all user tasks to be supported by the software and the important attributes of those tasks.

The documentation of the INITIATION phase should characterize the working environment in which the proposed software will operate. This would include a description of user tasks to be supported, the present methods and computer aids for carrying out these tasks, and the present limitations that the proposed system would overcome. The description should be oriented for the users' review, so that they may identify omissions or shortcomings of the proposed software solution.

*Maintain a continuing interaction with the users and their managers during the development period.

The potential users or their managers are authorities on the working environment in which the proposed system must function. They can provide vital information on the acceptable performance and design goals for the system, such as input data characteristics, tolerable response time, etc. Their strong support and involvement increases the likelihood of successful planning and design, and helps in obtaining needed requirements data and evaluations of the proposed system. A highly recommended approach is to have selected users serve as members of the project team, not necessarily to perform design work, but to help to decide and to document requirements. Project milestones should include scheduled review sessions as the requirements and alternative solutions are evolved, to obtain user comment and concurrence.

*Straightfoward automation of existing manual systems may not yield the greatest benefits from computers.

The computer's capability to do complex processing of large volumes of data at high speed generally means that computers can provide better analytical assistance to human decision-makers than could any number of clerical people. The way in which data records are organized for manual handling also may not be the best to use in computer storage and retrieval. The INITIATION, DEFINITION and DESIGN phases should include research toward new algorithms and data handling procedures that allow the computer to serve as more than a high-speed clerk and file cabinet. For example, there are mathematical procedures that can greatly improve inventory controls, scheduling for scarce resources, and plans for performance of complex tasks. Also, there are well proven computer techniques that provide rapid retrieval of the very data needed for a human task, rather than requiring people to manually hunt through long computer print-outs of all related data. Improved algorithms, carefully chosen for each application, can increase human and computer productivity by many times.

*Radical advances in automation carry exceptional risk--experiment with new approaches through prototype software, and be prepared to throw away one or more systems before confirming the design.

Too often, the cause for software management failure is overambition or executive zeal to bring the benefits of automation to a pedestrian manual operation in one great leap forward. Never undertake to produce in a single project step a system for which there is no technological precedent. Keep the scope of systems within the bounds of your experience with successful projects--advance your system capability in modest increments.

*Use consultants and visit other organizations to identify potential problems in building a system and to verify that system objectives are attainable.

Systems that have any possibility of challenging the state-of-the-art must be planned cautiously. Researchers, consultants, and organizations who have pioneered an innovative type of system can help define the major pitfalls and how they may be avoided.

*Systematically evaluate available software of a comparable type, whether commercial or other-user developed, before making a firm commitment to develop a new system.

The feature analysis technique is very useful for evaluating available software, and comparing proposed system requirements against the current state-of- the-art. Feature analysis proceeds by decomposing the requirements into individual functions and attributes, and then developing short descriptions of what the available software provides for each feature. The descriptions should be organized in a table for easy comparison. Available software, if adequate to the need, can usually be purchased for much less than the cost of producing similar programs. Even if new development is required, existing software may serve for some components and reduce overall system cost.

*Formulate and evaluate alternative solutions that differ in the risk and cost of development.

Prudent management requires a look at other choices before deciding upon one system to be bought or built. Since software cost and risk are now so important, they naturally are key considerations in conceiving of competing approaches. The possibility of meeting the need through commercial computing services or other sources of available software certainly should be presented as one alternative, it feasible. No definite guidelines can be offered on how to formulate alternatives. Clearly, thorough knowledge of the state-of-the-art is essential and will indicate candidate solutions that are more or less innovative.

*Estimate the future system costs by several techniques and adopt a conservative projection.

There is no well formulated cost estimation method that is highly favored and widely used. Various published data exist on programmer production rates, but different variables are involved and some important factors are omitted in one case or another. BROOKS (see References) gives a concise summary of published data. YOURDON and WALSTON also have data showing the gains possible with recent innovations such as structured programming. Experience and judgement, however, are the most accepted means to reach an overall recommendation on the time and funding needed. Published data should certainly be used as a guide, noting carefully the pertinent factors included. Examine costs of comparable systems and advertised costs of commercial packages for added evidence. Do a detailed cost breakdown of the system by major components and by major technical activities or work processes. Include operation and development costs, to have a full life cycle investment comparison of alternative systems. Above all, recognize the uncertainty of any early estimate of cost, and choose a conservatively high, yet reasonable figure. Reevaluate the cost estimate periodically as the development proceeds and better definition of the software evolves.

*Have the intended users review and approve the requirements and recommended solution defined in the INITIATION phase.

User concurrence, along with any evaluations and recommendations from the user review, should be in the record concluding the INITIATION phase. This could be as simple as initials in a project notebook for a very small project, or may require an official report and approval letter for a large project. To reach their assessment, users should be given a cost and benefit analysis of alternative systems, in addition to the general specification of each.

*Resist generalizing the system capability to accommodate requirements that cannot be resolved before DESIGN begins.

If an imposed schedule will require starting DESIGN before all requirements are defined, a judgement must be made of whether the uncertainty allows useful effort to proceed. If it does, then the unknown requirements must be anticipated and the most likely possibilities defined. The impact of these alternatives on the system design must be determined, and if great enough, it may be necessary to proceed with definition and design of several systems until the unknowns are eliminated. Be cautious about generalizing the required system capability to handle arbitrary, unknown demands. Generalized systems often involve great risk in development, and may have poorer performance and efficiency than more constrained and specialized designs. Be forceful in investigating potential needs, in order to resolve requirements questions and uncertainties as early as possible.

*Use engineering analysis, modeling or simulation, and experience data to establish the performance needed in critical software operations.

Software designers often assume that necessary performance can be attained by "tuning" programs once they are running. On the other hand, early decisions, for instance on the hardware, the programming language, or the principal algorithms to be used, could be so poor as to eliminate any hope of reaching a needed level of performance. Engineering analysis, coupled with measurements from comparable existing software, should be used to investigate the expected system performance as a function of the attainable performance for basic processing steps. Performance goals should be specified for the critical operations, to support feasibility judgements and to serve as validation criteria for the DESIGN and PROGRAMMING phases. Early decisions that constrain the design or implementation approach for the software should show clear evidence that the necessary performance would be achievable.

*Include with the Functional Requirements the performance and quality objectives that the delivered software must meet.

The specification of quality criteria should begin in the DEFINITION phase in order to have a basis for judging the subsequent design. Use the definitions given above as a starting point, to be expanded and specialized for a given project. For example, identify the system functions that are most subject to future modifications and the nature of the required modifiability. As another example, describe such useability characteristics as the processing points where recovery procedures are required and the acceptable form of recovery. Include the quantitative performance goals for principal algorithms and processing steps. In the Project Plan, define the technical reviews and the evaluation procedures that will be used to assure that the quality and performance objectives are met throughout the project.

•Develop the Functional Requirements in a structured format, to expedite review and validation.

A free-form narrative approach to documenting requirements will make it difficult, except for very simple programs, to isolate individual functions and performance figures for later confirmation of design. A better approach is first to decompose overall requirements into major functional groups, such as data entry or report generation, or into processing modes, such as online query or transaction auditing. Then describe individual requirements under each group with a separate statement of a function that must be provided or a performance or design attribute that must be achieved. Each requirement should include criteria for confirming its achievement, the testing or evaluation methods involved in confirmation, and any comments or discussion that amplify the requirement statement or show its origin.

•Do not conclude the DEFINITION phase without having an approved specification that incorporates changes and refinements emerging from user reviews.

Never proceed into detailed design when ambiguity or known errors exist in the requirements specifications. Basic disagreements are more likely to lead to fundamental redesign than to minor and easily rendered change.

•Formulate concrete, measurable milestones of progress for the design, programming, testing, installation, and initial operation of the system.

It is impossible to exercise control and take corrective action without concrete evidence of where problems lie. This means that progressive milestones must be conceived covering the life of a project. They must be events and accomplishments that are directly observable by the software manager, such as the receipt of a specified document, the successful operation of a given program, or the conclusion of a scheduled review meeting. They must be sufficiently frequent over time to serve as progress indicators or trouble alerts if not completed. The dependence of later milestones on earlier events should be revealed by drawing up a graph or network relating all milestones. Progress during DESIGN and PROGRAMMING should be measured by required events, such as reviews of specifications, and not by programmer estimates of percent of code completed. The accompanying chart describes a minimum recommended set of milestones over a system's life cycle.

•Establish controls and milestones that will identify and correct errors or omissions as early as possible.

The cost of correcting an error or adding capability to a system increases significantly as a project proceeds. By the time system testing begins, errors are over ten times more costly to fix than if they had been recognized in design. Additional time given to improving specifications and to thoroughly critizing them is worthwhile in reducing later project costs.

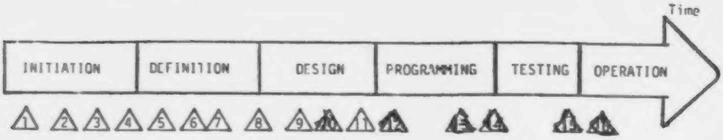
•Plan at least 10% of total project effort for management requirements and technical reviews.

When the needed effort has been underestimated, which too often is the case, careful reviews and management analysis may be thrown aside to put all effort into design and programming. Obviously this will work to the detriment of quality objectives, and probably will result in even more time required for testing and rework later. Make sure that project estimates include the time and effort for detailed review of work as it progresses, and for continuing review of project schedules and resource expenditures. Insist on using this time as intended, so that project status and problem sources can always be accurately defined.

•Plan contracts for design and production only after the detailed Functional Requirements and Project Plan have been prepared and approved.

Contractors cannot be selected or effectively monitored without the quality assurance and validation criteria that are embodied in these documents. They also provide the first practical basis for decomposing the development effort into work packages that different contractors might undertake. Contract requirements particularly should call for standardized documentation and coordinated format among the Functional Requirements and the succeeding design specifications, to aid in tracing and correlating results as the effort proceeds.

Figure 5. Chart of Recommended Minimum Milestones



Description of Milestones

1. Initiate Project Record with plan for feasibility study.
2. Complete analysis of user tasks to be supported.
3. Complete definition of alternative software solutions.
4. Complete feasibility study report with recommended solution.
5. Obtain user concurrence with recommended solution; budget approval for project.
6. Complete Functional Requirements.
7. Complete Project Plan.
8. Complete user validation of Functional Requirements.
9. Complete software architecture and definition of major modules.
10. Complete Test Plan, and Design/Coding Standards.
11. Complete Design Specification of all component programs; tools definition.
12. Complete validation of Design Specification.
13. Complete unit testing; accept all modules for source code control.
14. Initiate fault reporting procedures.
15. Complete integration testing.
16. Complete delivery and shakedown.

Milestones involving testing

GUIDELINES FOR DESIGN AND PROGRAMMING

*The second major source of problems lies in the quality of programs as they are produced.

About 40% of software errors are traceable to misinterpretations of specifications, errors in designing to them, or mistakes in writing program statements. Recent research to improve software management has concentrated on these problems. A variety of new techniques, such as structured programming and automated quality control, are beginning to come into widespread practice, but much more must be done. Misinterpretation of specifications would be reduced by more formal or mathematical specification methods rather than exclusive use of narrative English. Analytical methods of design, replacing the subjective criteria used heretofore, would reduce the frequency of programming errors and costly maintenance. Software managers must keep current with such developing areas in software engineering research. Continuing training programs, special in-house workshops, and experimental projects are good avenues for stimulating the computer staff toward improving their working practices and technical knowledge.

*On beginning the DESIGN phase, prepare an explicit specification of the quality design and programming standards that must be observed in the DESIGN and PROGRAMMING phases.

The Functional Requirements should be accompanied by definitions of mandatory quality objectives, that serve as a starting point. Using these, a list of design attributes can be specified, extending in detail to conventions on the use of programming language features and constraints for individual program design. This expansion of quality specifications therefore will encompass the Coding Standards to be used in the project, and will provide a framework for starting design of individual modules. For example, the quality standards may define the extent of diagnostic messages to be provided, the types of data errors to be detected by validation algorithms, the manner of isolating particular processes subject to future modification, etc.

*Develop the Design Specifications as a hierarchy of processes, specifying the input and output of each process, and the data flow interconnections.

A hierarchy is an effective means to decompose functional requirements into successively more detailed operations. The approach is well matched to what is called "top-down" design for software, which proceeds by successive refinements of general functions, until arriving at the component modules or programs that must be implemented to build the system. Documentation for this functional design approach is rather simple, and it portrays the overall system operation in an easily followed graphical way. The References include books and reports on the technique.

*Complete the Design Specifications before beginning to code.

It is vital that the design be completely worked out before programming begins, so that programming effort is not wasted on erroneous design notions and so that programming, when started, has a complete guide to assure quality work.

*Decompose the system into program modules that are simple and easily understood, usually meaning about 100 program statements in size.

The advantages of small modules are many. Their functions are more clearly and simply describable. This helps direct the programmer toward a clear objective. Their brevity improves clarity and speeds review for validity and quality. Progress may be more accurately evaluated when the units of programming work are small and readily understood.

*Define +track general service modules to reduce duplication in program modules.

In designing the system into modules, keep track of replicated functions and define modules that provide such functions as services that all programs may use. These service modules usually are those that manipulate standard data structures in a system of programs, for example, entering items in tables, retrieving items, managing storage allocation, etc.

*Use standard high level programming languages.

The use of vendor-unique programming language, especially machine or assembly language, should be prevented, except for very small portions of code where assembly language may be essential to achieving the performance goals of the software. These exceptional cases should be strictly controlled by explicit approval. Keep a standard high level language version of assembly programs for documentation and maintenance assistance.

*Use structured programming techniques.

Structured programming is a concept that encompasses programming team management, design methods, and programming technology. Numerous books and articles are available, describing the design and programming techniques especially. Published results from evaluation projects confirm that increases in programmer productivity and reductions in program errors are achieved, often as factors of two or greater. Investment in the training and management orientation needed to practice these methods will be worthwhile.

*Identify or develop programming aids to increase team productivity, and issue guidelines to the team for using them.

Specific recommendations on programming tools are given later. Selection of tools and specifications for unique tools needed for the project should be done as early as possible in the DEFINITION and DESIGN phases.

*Conduct weekly reviews by the programming team of ndividuals' work.

This practice helps all team members to keep informed of the design and production as it proceeds, and may act also as a training opportunity. But primarily it focuses the combined talent of the team on the quality of the team output as a whole. Errors or inferior design can be more quickly exposed than with reviews by a single manager. The results of each review session should be documented as a guide for the individual programmer to improve or correct his work. In structured programming, these review sessions are called "structured walk-throughs", to suggest that each programmer leads the team through the design and operation of his module in the same pattern as it was developed.

*Design and program for quality before performance.

Before programmers make any effort to tune their modules for improved performance, all quality characteristics required should be realized. Necessary comments and program arrangements for clarity, maintainability, modifiability, generality, and all required functions should be completed and confirmed in reviews.

*For large projects, conduct inspections at successive tages of design and production, and make all necessary anges before work moves to the next stage.

Independent & Review

In addition to team reviews of each programmer's progress, more formal inspections are recommended for large projects, to examine subsystems or closely related modules produced by different programming teams. These inspections should serve as progress milestones in the Project Plan. The inspection team may include experts outside of the teams, in addition to the team chiefs and the responsible programmers. The inspection team leader (or moderator) should not have been a participant in producing the subsystem under review and must have the authority to inform higher management of negative findings. A team size of about ten persons maximum is advisable, and separate reviews should be made when design is complete, when code and unit tests are complete, and when first integration tests are complete.

*Use iterative review techniques for validation.

Review & Testing

Validation, also sometimes called verification, means confirmation that a system is reliable, and meets its specifications as well as the requirements of the user environment. Team reviews or inspections should always trace design specifications back to functional requirements, taking note of findings from past reviews. In this way, the logical thread that stems from the original specification is followed to the working software, and successively examined for continuity and possible shortcomings. Reviews should evaluate the suitability of the software for the intended use, and whether the necessary quality is being achieved.

*In the PROGRAMMING phase, the individual programmer performs unit testing.

Unit testing determines that the individual programs work in isolation from others and meet the specifications given with the assignment. Unit testing must be performed according to standard criteria, rather than ad hoc methods contrived by each programmer. The programmer provides test reports for use in review and inspection tasks. Testing criteria are described in the guidelines for TESTING, which follow.

*Put program modules under source code control when unit testing is completed.

Completion of unit testing signifies internal project delivery of a module for integration with others in the overall software system. Source code control restricts further program changes to formally approved changes that have been decided from an overall system viewpoint as necessary to meet project goals. Source code control is needed for best use of programming effort in the terminal period of the development schedule.

*Before the DESIGN phase is complete, a formal test plan for the system should be approved.

[illegible]

The Test Plan should be started in the DEFINITION phase, for the Project Plan must account for the effort required for the testing strategy to be used. Some details on the number of tests, test inputs, etc. cannot be specified until the design is nearly complete. The Test Plan describes the TESTING phase effort and the testing schedule, and defines the standard test set, i. e. the complete range of test cases that will be repeatedly used to assure reliability of the delivered software. The development of this standard test set is a significant effort that must be completed by the start of TESTING. The test cases must be provided by manual calculations or other independent means, so that the correct output expected is known beforehand. Other means may include program analyzers, simulations, tables of random numbers or function values, results of comparable systems, etc. The depth of the Test Plan is indicative of the reliability the user may expect if testing is successfully completed. Ideally, the Test Plan is developed by a technical group that is independent of the designers and programmers, and that can require tests that focus on perceived weak points in the design. Users should be represented in test planning, and the final plan should be jointly agreed between users and the chief designer.

GUIDELINES FOR TESTING

*Delivered software for large systems typically has errors at a rate of one in every 300 program statements.

When TESTING begins, the errors present may be three or more times higher than this figure. Testing is the process of executing programs with representative input data or conditions, for which the correct results are known, to determine whether incorrect results occur. Testing provides information that helps to isolate errors. Debugging is the subsequent work of a programmer to explicitly identify errors and to correct them by changes to program instructions or data. Many large program systems never become error-free, i. e. "correct". Instead, they are maintained and managed in terms of the number of known errors that are allowed to exist--i.e. successive versions or releases are planned to reduce the known errors to some acceptable, but perhaps still high figure.

*Testing is the only practical way to detect program errors so that they may be eliminated.

Testing only determines if errors are present. It cannot verify that no errors exist in a program. Such confirmation is conceivable in only two ways. One is to exhaustively test the program on every possible input. Even for simple programs, this could take millions of years even if done automatically by a computer. The second conceivable way is to prove correctness as a logical and mathematical theorem about the program instructions. This approach is a current research topic in computer science, and proofs have been done manually for simple programs up to a few hundred instructions. Research work is now aimed at simplifying proof requirements through programming practices suited to this objective, and through computer theorem-proving software that would reduce the complex mathematical effort involved. No break-through is predicted in the near future that would make proof-of-correctness practical and economical for the average programmer to use on normally complex programs.

*Adopt specific criteria to govern unit testing of all programs.

Many different testing criteria are conceivable, but most have no direct relationship to possible errors and design flaws, and so provide no information to improve reliability. For example, one could require a specific number of test runs per program. But unless each test was appropriately distinct, no more benefit would be obtained than from running only one test. A random sample among most of a program's input values may be an effective criterion. But it is useful only if the valid input values are small in number, so that a significantly large number of random tests can be generated and run within the time and resources available. Other possible criteria are:

a) test every instruction at least once;

b) test every branch of the program;

c) test every logical path within the program with at most one iteration of every loop.

(A logical path is a sequence of instructions, some possibly repeated, from input to termination of a program.) The above three criteria are successively more stringent; they are preferable to many others because they are well directed at possible faults, and tend to be efficient in using test effort.

Recommended testing guidelines that are broadly applicable to software cannot be simply stated. The subject is technically complex and beyond the scope of this guide. The National Bureau of Standards is currently preparing a guideline that will include results from current research to develop testing theory and to lay down broadly effective principles. The most important consideration at this time is to avoid arbitrarily chosen tests by each programmer, that do not clearly exercise previously untested functions and logic.

*Schedule progressive tests to build up to a representative full system test.

Testing should begin with discrete tests of basic system functions that are needed in order to do more complex and realistic tests. Simple input, output, and control functions are examples. Keeping each test restricted to a given function allows the results to clearly identify problems encountered. Ultimately, the software should be exercised by composite test cases that represent full system operation and expected usage conditions. Composite tests should be conceived to stress the software performance. The limits on such factors as accuracy, acceptable volume of data, or number of concurrent users, should be verified if included in the specifications. The diagnostic features of the software should be exercised as well with test cases having invalid input. The Test Plan should encompass most, if not all, of the capabilities defined in the Functional Requirements.

*Use program analyzers to assure that all program functions have been exercised.

A program analyzer is a computer program that collects data about another program, including data about it as it operates. In particular, an analyzer can determine whether test inputs have caused all the program instructions to be used. This verifies that all capabilities have been exercised, and identifies extraneous code that may have no purpose and should be removed.

*Experience indicates that testing usually requires about 40% of available development time.

Careful, disciplined design should reduce the need for extensive testing, but more theoretical development and experience with improved techniques will be needed to assure this is always so. Certainly, better specifications would eliminate rework that sometimes is initiated during TESTING when it becomes clear that the design was incomplete or haphazardly contrived. But, it seems best to be conservative in project schedules as well as cost estimates, so allow a major portion of project time for testing and design rework.

*Use a fault reporting process to manage debugging and testing.

It is good practice to have a formal report system by which detected errors and discrepancies are recorded and fully described. These reports will help to confirm that all known errors are fixed before delivery. They also help to trace multiple instances of the same anomalous behavior, so that debugging assignments can be made for related problems and the debugging effort reduced.

*Conduct regression tests after program rework has been done.

The TESTING phase is primarily concerned with integration testing, i.e. tests to determine whether different program modules, produced rather independently, will work together correctly. When errors are found and eliminated by debugging, new undetected errors may be introduced. Regression testing means that previously successful tests are rerun to detect any introduced errors. A complete, successful run of the entire standard test set should be accomplished on the software before concluding TESTING and delivery to the customer.

*Supplement integration testing with system validation reviews.

Although specifications are expected to be of high caliber when PROGRAMMING begins, practical use of the resulting system during TESTING may reveal that original notions of desired operation were fallacious. Reviews of test sessions should consider the general suitability of the system as well as strict adherence to discrete functional specifications. Any discrepancies should be considered for rework, rather than delivering a system that in the judgement of users and designers will not be adequate for the user's purpose.

*Testing should assure a high measure of reliability that would be expected in operation of computer programs.

Correctness is the desired quality for software, but also is an ideal condition that often is not economically attainable or confirmable for complex software. So, the term "reliability" is useful to denote the lack of detected errors with respect to some basis of observation. Observations could be made over a long time period of repeated use of the software, and reliability then defined as the ratio F/N, with F the number of detected faults, and N the number of runs or trials of the program.

Software reliability measurements will reflect how the software is being used, as well as the faults or errors it contains. If routine usage is limited to previously debugged functions, reliability will apparently be high. If usage changes, reliability may suddenly degrade seriously, as previously unexercised code is used and its errors revealed.

Reliability measurement is a recent subject of software engineering research. More results are needed before a minimum acceptable figure can be recommended. The most effective use of reliability data in project management also is not certain until more project experience is at hand.

*Plan a shakedown period after delivered software is installed.

A shakedown period may be dictated by hardware alone if a new computer system is involved, but this kind of trial use may also effectively supplement the previously performed integration testing of software. The shakedown scenario should be as close to the actual anticipated usage as costs and procedures permit, but do not depend upon the shakedown operation for any operational service. Use the fault reporting process and records of shakedown run time to measure and evaluate reliability. The shakedown period needed to attain stable and reliable operation may be several months, particularly if hardware installation is extended over this time and considerable user training is being done.

*use the management techniques for development during the OPERATION phase as well.

The programming and design effort available during OPERATION is generally reduced compared with the development phases. However, the goals of software management do not change; indeed, maintenance costs may dominate the total life cycle cost of a software product. Maintenance work on software is still a design challenge, but one constrained by the program structure and instructions already in use. Unless technically controlled, maintenance may degrade reliability and other qualities that were initially present. Maintenance tends to disorganize and complicate the program structure, increasing the potential for errors and the cost of future changes. It should not be assumed that the existing software is a sufficient constraint to prevent unsupervised maintenance from becoming costly or haphazard. Maintenance should be conducted under a tight management regime that includes definition and scheduling of manageable work assignments, preparation and review of specifications, team review of design and code, standard tests, and documentation.

*keep a failure and discrepancy reporting process to continually manage software reliability.

Failure reports are normal in the TESTING phase and the initial period of OPERATION, but they continue to be important as maintenance continues over the years. These reports are needed for software managers to accurately determine if maintenance work is being accomplished as assigned and if greater testing and rework effort is warranted. Be cautious about proposed improvements that are not strongly supported by users or indicated by problem reports.

*Evaluate software periodically and plan improvements to avoid obsolescence and to minimize future conversion costs.

The high investment in current software and the high cost of replacement often lead to prolonged use of software designs that have become obsolete. Obsolescence may be due to changing technology or economic factors, or to changes in organizations and work loads. Major software subsystems should be evaluated periodically, to identify modifications that would eliminate operational limitations, or would use improved hardware or software technology. Major improvement in an existing system may be economically done through progressive stages of rework. Equally important, rework to standardize or restructure component routines may increase the feasibility of transfering the software directly to a future computer configuration. Such prospects emphasize the importance of standard documentation, because the replacement of modules or interfaces would be impractical without clear definition of the existing software design.

*use configuration management methods to control software status.

Configuration management involves close control of operational software modifications, to assure that continuing service is not adversely affected by changes and that the changes made are sufficiently important to justify their cost within the total maintenance effort available. Proposed changes should be agreed to jointly by the programming group and the users, and scheduled with the appropriate priority as part of the total workload of the programming team. Changes should first be made to a test copy of the software, not the same copy used for everyday service. Thus testing can proceed without delaying operational service, and only a fully checked out system is delivered for use.

*Periodically review the Project Record and current documentation, to insure that maintenance results are being recorded for later use by other programmers.

Maintenance responsibility is sometimes misconstrued as a personal prerogative that may be done with each programmer's style, unless management exerts its authority for quality control to meet the organization's needs. Documentation is a tedious but crucial chore, and it is inexcusable for operational software to be altered with no evidence except in the code itself. As in all phases of software work, management action should be as immediate as possible. The team review process is recommended for it may remove any appearance of management inquisition. Weekly meetings that include reviews of documentation updates may be sufficient to keep project information current and complete.

THE NEED FOR SOFTWARE PRODUCTION TOOLS

*Software tools are computer programs that automate tasks in the management, production, and testing of software.

A wide variety of tools have been developed and used in the software field, but few standard tools are recognized as applicable to every project. The most widely used tools are the compilers for high level programming languages, that transform program statements into executable machine codes. Many other tools can appreciably reduce the tedium of programming tasks and can expedite the application of standards to a programmer's products. Surveys of computer installations, as well as recent project experiences, show that the value of effective tools is not widely appreciated, and an adequate repertoire of effective tools is seldom provided. The importance of good software production tools is confirmed overwhelmingly by all recent experience and authoritative research in software engineering.

*Provide interactive computer support for programming on every project.

Developing programs and documentation on an interactive computer facility significantly improves the efficiency and working conditions for the programming team. Programmers can concentrate on technical problem solving without being limited and burdened by scheduling of computer runs, as in a batch computer support facility. Tentative design ideas can be experimentally programmed and evaluated in a short time, allowing the development of a quality design to move along rapidly. Various debugging and testing tools can be used to shorten the time for identifying sources of errors. A separate interactive support facility is advisable when the operational computer is unsuitable for interactive usage because it is intended for a batch or real-time application. The technology and economics of interactive support software have been amply proven, and the availability of basic systems is adequate for this general recommendation.

*Use preprocessors to supplement a standard high level programming language.

A preprocessor accepts programming statements written in an improved dialect of a standard programming language, and translates them into standard language acceptable to a compiler program. The merit of a preprocessor is that restrictive conventions that still exist in standard languages, such as card column orientation, can be circumvented for improving programmer productivity and reducing errors, without losing the transportability advantages of the standard language. For example, structured programming could be applied even with FORTRAN, which is inherently unsuited for it. Equally important, many of the coding standards of a project could be checked for each program by a preprocessor, as a means of automated quality control. However, a preprocessor may be difficult to use together with a symbolic debugger, because the debugger output would refer to the preprocessor output statements, and so debugging results would be difficult to correlate with the source program statements.

•Establish a standard library of specialized software tools.

Software tools generally are inexpensive, often available from other users for the cost of reproduction, and sometimes concise and easily modified for unique project needs. Tools are usually specific to one programming language, and they perform a limited set of functions on programs written in that language. For each language in use then, a basic set of tools can be assembled, with each one assisting in an important, distinct design and programming task.

•Use an editor, program librarian, and program analyzer as standard tools.

An editor assists the programmer in making numerous changes and additions to a program, without the burden of filling out coding forms or manipulating format details, and with less risk of errors in repetitive, trivial changes. A good editor will automatically perform routine formatting and standardization actions, and with prompting and checking facilities included, it can prevent or identify trivial syntactic errors that a programmer may make.

A program librarian is a necessary complement to an editor, and provides for storing on the computer and retrieving all program modules and distinct versions of them that may develop during a project. The usual file system of an interactive computer facility may serve as an adequate program librarian. Features that are useful and often provided in special tools for this purpose include capabilities to time and date stamp different modules, to employ special naming conventions for programs, to efficiently store related program versions by storing only the relative changes, etc.

A program analyzer is software that scans the source statements in another program to collect information about them or to provide for data collection about the program behavior when it is executed. A variety of analyzers may be useful, for special formatting of programs, for auditing adherence to coding standards, for extracting cross reference and data usage reports, for performance measurement, for test monitoring, and similar purposes.

Many tools of these types are available, and special tools are often economical to develop for a project.

*Use a simple project information system to maintain data on schedules, milestones, and resources used on distinct tasks and activities.

Simplicity is the key to keeping the management and administrative effort within bounds for small projects. For large projects, fiscal and contractual requirements may dictate extensive record keeping, but it is still advisable for technical management to have simplified project data that may expose basic problems, rather than to be buried under all possible details. If a computer information system is used, it should maintain a gross network of major milestones and produce a few brief reports on manpower and other expenditures for these milestones and other support activities.

STANDARDS FOR QUALITY CONTROL

A number of distinct quality control techniques are known to be effective, and some have been briefly described in the preceding. In all cases, there is no documented standard of practice for the technique, but there are published articles and sources of experience to help in developing appropriate standards for an interested organization. A summary definition of known techniques is given below.

PROJECT RECORD--An official journal of project plans, specifications, schedules, work assignments, budgets, technical standards, etc. that serves to inform all concerned parties of the goals, work methods, history, and status of the effort.

PROJECT PLAN--A detailed plan of milestones, resource allocations, and quality control procedures to be used in the development phases for a software product. Incorporated in the Project Record and updated as needed during development.

DOCUMENTATION STANDARDS--Uniform requirements on documents for the project and the software, that define scope and identification of each document, as well as format and content, and that cover the principal specifications and technical descriptions, both internal to the project and deliverable with the software.

AUTOMATED DOCUMENTATION AIDS--Computer software for text editing and document formatting, and for automated preparation of program documentation such as flow charts or cross reference listings of data elements and program statements.

STRUCTURED SPECIFICATIONS--The approach of coordinating all specification documents in format and content, and using formal or mathematical specification techniques to reduce ambiguity. This facilitates successive validation milestones as increasingly detailed specifications are developed. The series of specifications involved begins with the feasibility study description of system concept, continues with the functional requirements, and extends to the individual program module specifications.

QUALITY and PERFORMANCE REQUIREMENTS--Augmentation of the functional requirements, to state the specific quality attributes expected of the software and the quantitative performance necessary in key operations or test cases.

SPECIFICATION CONTROL--A formal process of reviewing and approving proposed changes to specifications, with consideration of their technical validity, impact on project resources and schedule, and impact on work completed or in progress. The goal is to inhibit continuing, capricious changes, and instead stabilize the specifications on distinct deliverable versions of the software that can be managed for production.

STRUCTURED PROGRAMMING--Technical practices and criteria governing particularly the programming techniques and language usage for increasing programming productivity and reducing program errors.

TOP-DOWN PROGRAM DEVELOPMENT--The approach of implementing and testing program systems by building program modules starting with those at the most general, application-oriented level (the "top"), and proceeding successively down to the most specialized, computer-dependent level (the "bottom").

DESIGN and CODING STANDARDS--Detailed specifications of generally applicable design requirements based on quality objectives, including such programming conventions as indenting and spacing of program statements, use of comments, naming of variables, use of language features, etc.

SOURCE CODE CONTROL--The process of labeling and archiving program modules accepted for formal technical control, thereby limiting future programmer modifications to formally reviewed and approved changes.

TEAM REVIEWS--Technical team conferences to audit design proposals, specifications, and completed programs for quality.

UNIT FOLDERS--The programmer's technical records of design and testing work on individual program modules, sometimes called "unit development folders". As standard documents in a project, these augment the project record and specifications by providing more technical documentation as necessary for reviews and inspections. Especially important in large projects subject to frequent personnel changes or reassignments.

INSPECTIONS--Formal review and auditing of subsystems or groups of program modules at design, code, and test stages, by an independent technical team. Applicable to larger projects.

STANDARD PRODUCTION TOOLS--Computer software to aid the programming team in creation, documentation, testing, and analysis of the deliverable programs. These would include, for example, the standard programming language compiler, program librarian, test data generator, symbolic debugger, etc.

PROGRAM ANALYZERS--Software tools for automated analysis of program modules, to confirm complete adherence to coding standards and quality criteria, and to assure thorough testing according to standards.

PROJECT INFORMATION--Recording and analysis of the allocation and expenditure of team effort on different modules and milestones for the program system, and on various production activities, in order to monitor progress and to support best practical resource management.

TESTING STANDARDS--Specific criteria governing the program testing to be performed, to assure that programs are uniformly tested by all programmers and that all program instructions and capabilities are tested to the extent practical.

INDEPENDENT TEST PLANS--use of independent technical specialists to prepare comprehensive tests of the integrated software system.

FAULT REPORTING--A formal process for documenting observed errors or discrepancies in program operation and capability, to aid debugging and reliability assessment, and to validate software design.

CONFIGURATION MANAGEMENT--A formal process of controlling the operational software status and reviewing proposed improvements, to best direct maintenance effort and to minimize impact of maintenance and testing on operational service.

FITTING CONTROLS TO THE PROJECT SIZE

Software projects differ tremendously in complexity, acceptable schedule, and available resources. More than anything, the available personnel determine what technical goals are achievable and what quality controls may reasonably be imposed. Figure 6 illustrates a basic judgement of where the above controls are best utilized in regard to project size. The controls that are perceived to be more demanding of technical or supporting effort are not recommended for smaller projects.

Figure 6. Chart of Recommended Quality Controls

Quality Control	Development or Improvement Effort in Man-Years			
	Less than 1/2	1/2 to 2	2 to 5	Over 5
Project Record	X	X	X	X
Project Plan	X	X	X	X
Quality/Performance Requirements	X	X	X	X
Standard Documentation	X	X	X	X
Program Analyzers	X	X	X	X
Standard Production Tools	X	X	X	X
Structured Programming	X	X	X	X
Top-Down Development	X	X	X	X
Structured Specifications		X	X	X
Specification Control		X	X	X
Automated Documentation Aids		X	X	X
Design/Coding Standards		X	X	X
Team Reviews		X	X	X
Testing Standards		X	X	X
Project Information		X	X	X
Source Code Control			X	X
Unit Records			X	X
Fault Reporting			X	X
Configuration Management			X	X
Inspections				X
Independent Test Plans				X

FINAL HOMILIES

A few simple cautions may help to avoid unqualified failure in undertaking software development, given present knowledge of how best to do it.

Choose reasonable goals. Although computer technology has amazing capabilities, we remain primarily limited by human creativity in regard to software achievements.

Have a detailed plan. Important tasks may be forgotten or expectations may expand unless all is recorded as a reminder.

Put one person in control. There should be strong leadership toward one well understood concept.

Use good tools. Productive tools are the only certain way to accelerate human accomplishments.

Cater to the software user. The user judgement is the ultimate answer on success or failure of the product.

Build up in steps. Confidence from successful achievement of modest goals will lead more quickly to grander objectives.

Admit mistakes early. Pursuit of the hopeless leads only to further despair--poor choices and misconceptions require drastic remedies as early as possible.

Consider starting over. Patching a poor result has limited return; a wiser action may be to start fresh at the beginning.

REFERENCES

Baker, F. T., "Structured Programming in a Production Programming Environment", *Proceedings of the International Conference on Reliable Software*, 21-23 April 1975, Los Angeles, pp172-185, IEEE Computer Society, Long Beach, California.

Brooks, F. P., Jr., *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley Publishing Co., Reading, Mass., 1975, 195p.

Comptroller General of the United States, *Lessons Learned About Acquiring Financial Management and Other Information Systems*, August 1976, 61p, available from U. S. Government Printing Office, Washington, D. C. 20402, Stock No. 020-000-00136-1.

Endres, A., "An Analysis of Errors and Their Causes in System Programs", *IEEE Transactions on Software Engineering*, v. SE-1, n. 2, June 1975, 140-149, IEEE Computer Society, Long Beach, Calif.

Fagan, M. E., "Design and Code Inspections to Reduce Errors in Program Development", *IBM Systems Journal*, v. 15, n. 3, 1976, 182-211.

Hecht, Herbert, *Measurement, Estimation, and Prediction of Software Reliability*, NASA CR-145135, Aerospace Corporation, El Segundo, Calif., January 1977, 41p.

IBM Corp., Data Processing Division, *Improved Programming Technologies--An Overview*, Installation Management Report GC20-1850-0, October 1974, 19p.

IBM Corp., Data Processing Division, *HIPO--A Design Aid and Documentation Technique*, Installation Management Report GC20-1851-0, October 1974, 130p.

Kernighan, B. W. and Plauger, P. J., *The Elements of Programming Style*, McGraw-Hill Book Company, New York, 1974, 147p.

Kernighan, B. W. and Plauger, P. J., *Software Tools*, Addison-Wesley Publishing Co., Reading, Mass., 1976, 338p.

Mills, Harlan, "Software Development", *IEEE Transactions on Software Engineering*, December 1976, 265-273.

Myers, Glenford J., *Software Reliability: Principles and Practices*, John Wiley and Sons, New York, 1976, 360p.

National Bureau of Standards, *Guidelines for Documentation of Computer Programs and Automated Data Systems*, Federal Information Processing Standards Publication 38, 1976 February 15, 55p.

Rochkind, M. J., "The Source Code Control System", *Proceedings of the First National Conference on Software Engineering*, 11-12 September, 1975, Washington, D. C., pp37-43, IEEE Computer Society, Long Beach, Calif.

Rubey, R. J., et al., "Quantitative Aspects of Software Validation", *IEEE Transactions on Software Engineering*, v. SE-1, n.2, June 1975, 150-155, IEEE Computer Society, Long Beach, Calif.

Van Tassel, D., *Program Style, Design, Efficiency, Debugging, and Testing*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974, 256p.

Walston, C. E. and C. P. Felix, "A Method of Program Estimation and Measurement", *IBM Systems Journal*, *v*. *16*, *n*. *1*, *1977*, *54*-*73*.

Yourdon, E., *Techniques of Program Structure and Design*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975, 364p.

Yourdon, E., "A Case Study in Structured Programming: Redesign of a Payroll System", *Digest of Papers of IEEE COMPCON*, September 9-11, 1975, Washington, D. C., 119-122, IEEE Computer Society, Long Beach, Calif.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NO. NBS SP-500-11

2. Gov't Accession No.

3. Recipient's Accession No.

4. TITLE AND SUBTITLE

COMPUTER SCIENCE & TECHNOLOGY:

Computer Software Management: A Primer for Project Management and Quality Control

5. Publication Date July 1977

6. Performing Organization Code

7. AUTHOR(S) Dennis W. Fife

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)

Same as Item 9

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 77-608127

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

Today, providing computer software involves greater cost and risk than providing computer equipment, because hardware is mass produced by industry using proven technology, while software is still produced mostly by the craft of individual computer programmers. This brief guide is intended for managers who are responsible for computer projects, to explain the use of quality controls and software management methods. The typical problems of software development are explained. Over twenty distinct quality controls are defined, and recommendations are given for software management actions. Empirical information is included that would help top executives to appreciate the potential problems and importance of software management.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Computer management; computer programming; computer project control; computer software; software engineering; software quality; software reliability.

18. AVAILABILITY ☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☒ Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13.10:500-11

☐ Order From National Technical Information Service (NTIS) Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED

20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED

21. NO. OF PAGES 58

22. Price

USCOMM-DC 29042-P74

BEST COPY AVAILABLE

FIPS PUB 65

FEDERAL INFORMATION
PROCESSING STANDARDS PUBLICATION
1979 AUGUST 1

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

FIPS

Guideline
FOR AUTOMATIC DATA PROCESSING
[redacted]

Risk Analysis Pg 5-9

CATEGORY: ADP OPERATIONS
SUBCATEGORY: COMPUTER SECURITY

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, ***Secretary***
Luther H. Hodges, Jr., ***Under Secretary***
Jordan J. Baruch, ***Assistant Secretary for Science and Technology***
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, ***Director***

Foreword

The Federal Information Processing Standards Publication Series of the National Bureau of Standards is the official publication relating to standards adopted and promulgated under the provisions of Public Law 89-306 (Brooks Act) and under Part 6 of Title 15, Code of Federal Regulations. These legislative and executive mandates have given the Secretary of Commerce important responsibilities for improving the utilization and management of computers and automatic data processing systems in the Federal Government. To carry out the Secretary's responsibilities, the NBS, through its Institute of Computer Sciences and Technology, provides leadership, technical guidance and coordination of Government efforts in the development of guidelines and standards in these areas.

The need for physical, administrative and technological measures to protect Federal agency data and computer installations has become an acknowledged fact. In order to assure the cost effective selection of such safeguards, a risk analysis is required by the Office of Management and Budget Circular A-71, Transmittal Memorandum No. 1. This Guideline for performing a risk analysis is made available by the NBS to assist Federal agencies in implementing the OMB requirements for the security of Federal automated information systems.

James H. Burrows, *Director*
Institute for Computer Sciences
and Technology

Abstract

This document presents a technique for conducting a risk analysis of an ADP facility and related assets. Risk analysis produces annual loss exposure values based on estimated costs and potential losses. The annual loss exposure values are fundamental to the cost effective selection of safeguards for the security of the facility. An ADP facility of a hypothetical government agency is used for an example. The characteristics and attributes of a computer system which must be known in order to perform a risk analysis are described and an example is given of the process of analyzing some of the assets, showing how the risk analysis can be handled.

Key words: ADP availability; annual loss exposure; application system vulnerability; computer security; data confidentiality; data integrity; data security; physical security; procedural security; risk analysis; risk assessment; systems security.

Nat. Bur. Stand. (U.S.), Fed. Info. Process. Stand. Publ. (FIPS PUB) 65, 27 pages.
(1979)
CODEN:FIPPAT

For sale by the National Technical Information Service,
U.S. Department of Commerce, Springfield, Virginia 22161.

Federal Information
Processing Standards Publication 65

1979 August 1

ANNOUNCING THE

GUIDELINE FOR AUTOMATIC DATA PROCESSING RISK ANALYSIS

Federal Information Processing Standards Publications are issued by the National Bureau of Standards pursuant to the Federal Property and Administrative Services Act of 1949, as amended, Public Law 89-306 (79 Stat. 1127), Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 Code of Federal Regulations (CFR).

Name of Guideline. Guideline for Automatic Data Processing Risk Analysis.

Category of Guideline. ADP Operations, Computer Security.

Explanation. This Guideline explains the reasons for performing a risk analysis, details the management involvement necessary and presents procedures and forms to be used for risk analysis and cost effective evaluation of safeguards.

Approving Authority. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Maintenance Agency. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Cross Index.

a. Federal Information Processing Standards Publication (FIPS PUB) 31, Guidelines for Automatic Data Processing Physical Security and Risk Management.
b. Federal Information Processing Standards Publication (FIPS PUB) 39, Glossary for Computer Systems Security.
c. Federal Information Processing Standards Publication (FIPS PUB) 41, Computer Security Guidelines for Implementing the Privacy Act of 1974.
d. Federal Information Processing Standards Publication (FIPS PUB) 46, Data Encryption Standard.
e. Federal Information Processing Standards Publication (FIPS PUB) 48, Guidelines on Evaluation of Techniques for Automated Personal Identification.

Applicability. This Guideline is applicable to all Federal agencies required to take action under the Office of Management and Budget Circular A-71, Transmittal Memorandum No. 1 of July 27, 1978, to ensure an adequate level of security for agency data.

Implementation. This Guideline should be referenced in the formulation of plans by Federal agencies for performing a risk analysis, whether or not the analysis is to be carried out by agency personnel or on contract.

Specifications. Federal Information Processing Standard 65 (FIPS PUB 65), Guideline for Automatic Data Processing Risk Analysis (affixed).

Qualifications. This Guideline has been prepared in order that a technique may be available for Federal agencies desiring to use it. However, it has become apparent that risk analysis technology is still in the evolutionary phase. As such, its further development would be seriously impeded by the establishment of a Federal risk analysis standard which required all agencies to adopt exactly the same methodology. Nevertheless, the needs of the Federal Government can only be met by the performance of risk analyses. Bearing in mind the pressure of both of these thrusts, the National Bureau of Standards is conducting an effort to identify the necessary constituent factors of risk analysis. With these established in a standard, Federal agencies will be able to conduct, or to have conducted for them, risk analyses with high confidence in the reliability of the product. On the other hand, research in the area will not be deterred by the inflexibility of an already prescribed methodology but should be encouraged by the setting of basic criteria and the challenge of developing and refining more sophisticated and more easily applied techniques.

Where to Obtain Copies of the Guideline. Copies of this publication are for sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22161. When ordering, refer to Federal Information Processing Standards Publication 65 (NBS-FIPS-PUB-65) and title. When microfiche is desired, this should be specified. Payment may be made by check, money order, or deposit account.

Acknowledgment

The procedure for risk analysis presented here is based on the work of Robert H. Courtney, Jr. of the IBM Corporation, who, while a member of the Federal Information Processing Standards Task Group on Computer Systems Security, gave his kind permission for its adaptation to the needs of the Federal Government.

Appreciation is also expressed to those other members of the Task Group who contributed innovative and imaginative ideas to the development of this Guideline.

1. INTRODUCTION

Hand in hand with the increase in awareness of the need for computer security has come the need for a method of quantifying the impact of potential threats on organizations supported by automatic data processing. [redacted]

There are any number of techniques for performing such analyses but two key elements must always be considered:

1. The damage which can result from an event of an unfavorable nature.
2. The likelihood of such an event occurring.

[redacted] It serves to point out the risks which exist; the required protective measures are then selected accordingly. An analysis shows the current security posture of ADP processing in an organization; it then assembles the basic facts necessary for the selection of adequate, cost effective safeguards. A secondary benefit of a risk analysis is the increased security awareness which will be apparent at all organizational levels, from management through operations.

A risk analysis provides management with information on which to base decisions, e.g., whether it is best to prevent the occurrence of a situation, to contain the effect it may have, or simply to recognize that a potential for loss exists. Because a risk analysis is the basis for such decisions, its estimates of loss or damage must be presented, where possible, in a quantitative, comparative fashion.

There are a number of other methods of inspecting, testing or evaluating the security of computer systems, such as penetration attempts, security audits, checklists and questionnaires. However, none of them can take the place of a risk analysis because their purposes are different and they do not consider the key elements of damage and likelihood of occurrence.

Risk analysis is not a task to be accomplished once for all time. It must be performed periodically in order to stay abreast of changes in mission, facilities and equipment. And since security measures designed at the inception of a system have generally proved to be more effective than those superimposed later, risk analysis should have a place in the design phase of every system.

The major resource required for a risk analysis is manpower—highly skilled manpower. For this reason the first analysis will be the most expensive, as subsequent ones can be based in part on previous work and the time required will decrease to some extent as expertise is gained.

The time allowed to accomplish the risk analysis should be compatible with its objectives. Large facilities with complex, multi-shift operations and many files will require more time to complete than single-shift, limited production facilities. If meaningful results are expected, management must be willing to commit the resources necessary for accomplishing this undertaking.

[redacted]

2.1 Management

[redacted]

1. [redacted]
2. [redacted]

3. [redacted] and formal delegation of authority and responsibility; and

4. [redacted]

Management should leave no doubt that it intends to rely on the findings of the risk analysis team. The scope of the project should be defined to encompass ADP users (this will probably include all departments and any users outside the organization) as well as the actual ADP facility, equipment and personnel.

2.2 Risk Analysis Team

The selection of the risk analysis team is critical to the outcome of the project. It is important to obtain representation from the organizational components responsible for the following:

- ADP operations management
- Systems programming (if separate from ADP operations)
- Internal auditing
- Physical security
- Data files under consideration
  (Very probably, all the applications processed by the facility will not be the responsibility of the same organizational component; in that case, a component need only be represented when its own data files are being considered.)
- Programming support of the files under consideration.

These entities should be represented on the team by people who are well informed both of their own component's mission and its relationship to the overall organizational mission. The task team leader should be equally knowledgeable and should come from one of the first three components listed above, but should not be that component's representative. In other words, the team leader should not wear two hats—one as leader and one as representative. None of this should be construed as precluding others from participation on the team and, certainly, departments such as legal and personnel should at least be consulted.

The leader and the team members should be designated in writing; their duties, responsibilities and any accompanying authority should be outlined. It should also be understood that the job cannot be done adequately if alternates are assigned. There may be a tendency on the part of the team members, in an effort to do a thorough job, to collect more information than is absolutely necessary; they should be cautioned about this as it can prolong the task.

There are reliable commercial firms which perform risk analyses on contract. It may be that management will decide to select one of them in preference to performing the task internally. That option should not be chosen in lieu of understanding the purpose and techniques of risk analysis, but rather in the interest of efficient resource utilization. The individuals who should serve on a risk analysis team are the same ones who will be needed to supply information to the contractor and to make certain that the product is a risk analysis rather than a list of vulnerabilities together with a list of intuitively chosen solutions. Although organization members could devote somewhat less time to it, especially in an organization maintaining a large number of applications systems and supporting files, they should still be readily available to the contractor throughout the risk analysis.

2.3 Allocation of Time

Risk analysis is a time-consuming process and one which cannot be hastened. Previous experience or a previous risk analysis to refer to will help considerably as will having all the necessary information readily available. At best, the consideration of each data set or file in the light of the hazards which beset a system is a tedious business, but one which should only be delegated to subordinates with great deliberation because of the level of knowledge and experience required in the decision process. It can be a very enlightening task, however, and one which may lead to system simplification.

The assignment of some individuals to the team may create a hardship for their organizational components, which will be forced to do without their services, as well as on the team members, who will feel compelled to rush through the risk analysis to get back to their

mally assigned duties. An agreement that
team will meet only half of each day would
viate some of these burdens.

2.4 Management Review

'op management should review both the pre-
nary findings and the final results of the risk analysis team for reasonableness, policy adherence and organizational unity before a protection plan is formulated. At the very least, the plan will require coordination with fiscal and administrative departments, and will probably be included in the organization's long-range planning.

3. PRELIMINARY SECURITY EXAMINATION

order to have a firm basis for conducting a
analysis, the team should initiate the proj-
by surveying the organization's existing
security, the cost of replacing assets, and
actual threats to which the organization's
processing is vulnerable. They will gain
/ledge from the survey which may some-
: reduce the amount of time required for
isk analysis. The natural inertia of getting
ed in such a group is easily overcome be-
e of the three specific products that are re-
d from this preliminary phase—the list of
s replacement costs, the list of threats to
h the facility is actually vulnerable, and
ist of existing security measures.

3.1 Asset Costs

e product of the examination should be a
f the replacement costs, or best estimates
of, of resources and facilities: the com-
(s), related equipment, data, buildings,
The total of all should be noted. Better
any other information available at this
this figure will give an indication of the
for security. If the risk analysis is being
in the system design phase, both the in-
d value of data in the completed system
he probable increase in the cost of acquir-
should be considered.

3.2 Threats

ther product of the preliminary phase
l be a list of the actual threats to which
DP facility and its resources are exposed. For instance, the occurrence of a tornado is a real possibility in the interior plains; in most coastal regions it is only a very remote possibility. Identifying the actual threats will give the risk analysis team a feel for the vulnerabilities, or possibilities for damage, of the facility and the systems they will be analyzing. Again, if the risk analysis is being done in the system design phase, an effort should be made not only to identify existing threats but to predict any future ones which might result from the implementation or operation of the system. The areas in the organization which should be surveyed for this purpose include:

- Personnel—hiring and termination procedures, scope and amount of training, quality of supervision at all levels.
- Physical Environment—neighborhood, quality and reliability of utilities, building design, operation and maintenance, physical access controls.
- Hardware/Software Systems—operational availability, change controls, software features, documentation.
- Data Communications — hardware and transmission circuits, procedures to validate and control distribution of messages.
- ADP Applications—technical design, documentation, standards. (Also see Appendix A.)
- Operations—standards and procedures for source document protection, information

dissemination, I/O control, tape library, forms, computer room processing, user interface, housekeeping and maintenance, production control, contingency planning.

Understanding the factors which contribute to system vulnerability is important in performing a risk analysis. These factors are hardly ever discrete and unrelated. Below is a sketchy example of the kind of approach which can be used to ferret out vulnerabilities.

- Natural Disasters. What kinds of natural disasters might reasonably be expected to occur? To what extent will the facility, processing availability, data, supplies, utilities, local transportation, etc., be affected?
- Environment. What special hazards such as explosives, flammable products, unused or unguarded buildings are nearby? What can be the aftermath of a fire in the vicinity? What is the proximity of the fire department?
- Facility Housing. Is ADP facility the sole occupant of the building? If not, what others? By whom is the building administered? By whom maintained? What construction is it? What warning devices and preventative equipment are installed? How close is it to heating equipment, cooking equipment, other fire hazards? What kind of floors and ceilings are there?
- Access. Is access to processing local or remote? Can an intruder gain access to processing, to data, to software, to equipment, to storage media, to preprinted forms, to supplies, to documentation, to output, to trash? Can an employee do the same? Accidentally? Maliciously? For profit?
- Work Scene. Is employee/management relationship satisfactory? How well do supervisors know personnel? Does management understand problems of personnel on shifts? How well do supervisors relay employee problems to management? Are employees loyal?
- Data Value. How much can an intruder gain by penetrating the system or disclosing data or disrupting operations? How much can a subject be hurt by unauthorized disclosure of data or by incorrect data? How much can the organization be hurt by disclosure of data or by basing decisions on incorrect data or by delayed processing availability?

3.3 Existing Security Measures

The last product of this phase should be a list of all security safeguards currently in effect, whether or not the original purpose of such features (e.g., storage media logs, control of printout distribution, data entry quality controls) was to protect. It will in fact be seen that good management practices generally promote security. Specific security measures, such as perimeter fences, guards, entrance badges, etc., may be for the protection of all offices and facilities in the building and would be in place even if the ADP facility were located elsewhere. The threats against which each of these in-place measures is specific should also be listed.

3.4 Management Review

The results of these surveys should be presented to management immediately upon completion. These results may point to the need for temporary safeguards until a final security plan, based on a complete risk analysis, can be placed in effect.

4. RISK ANALYSIS

Regardless of the cause, any harm which occurs in automatic data processing manifests itself as a loss to the organization of one, or more, of the following conditions:

DATA INTEGRITY—The state that exists when automated data is the same as that in the source documents, or has been correctly computed from source data, and has not been exposed to accidental alteration or destruction. Incomplete data, unauthorized changes or additions to the data, and erroneous source data are all considered violations of data integrity.

DATA CONFIDENTIALITY—The state that exists when data is held in confidence and is protected from unauthorized disclosure. Misuse of data by those authorized to use it for limited purposes only is also considered to be a violation of data confidentiality.

ADP AVAILABILITY—The state that exists when required ADP services can be performed within an acceptable time period even under adverse circumstances.

To prevent the risk analysis from bogging down in detail, the team should concentrate on the potential results of undesirable events, i.e., on the extent of the damage which they can cause, rather than on why they occur since the harmful events to which the organization is vulnerable have already been identified in the preliminary phase.

4.1 Elements

The essential elements of risk analysis are an assessment of the damage which can be caused by an unfavorable event and an estimate of how often such an event may happen in a period of time.

As it will be impossible for the team to know absolutely either the impact or frequency of many events, these must be estimated using a combination of historical data, the team's knowledge of the system, and their own experience and judgment. However, estimates within an order of magnitude are sufficiently accurate for the purpose of risk analysis in most cases. Later, at the time of selecting safeguards, if it becomes important to refine specific items, that can be done, but during the analysis gross statements of impact and frequency are all that are required.

4.2 Expressions of Impact and Frequency

Quantitative means of expressing both potential impact and estimated frequency of occurrence are necessary to performing a risk analysis.

To date no better common denominator has been found for quantifying the impact of an adverse circumstance—whether the damage is actual or abstract, the victim a person, a piece of equipment or a function—than monetary value. It is the recompense used by the courts to redress both physical damage and mental anguish. Some methodologies advocate the use of abstract symbols of impact. "$" is, in fact, a symbol, yet one which transfers directly to fiscal usage without any intermediate translation.

Since impact will be expressed monetarily and fiscal matters are organized on an annual basis in Federal agencies, a year is the most suitable time period to specify in expressing expected frequency of occurrence of threats. Some threats occur only once in a number of years while others happen many times a day. Such frequencies are not always easy to express in terms of years: "five times a day," for instance, converts to "1825 times a year" and "once every five years" converts to "one-fifth of an occurrence per year."

The time needed for the analysis will be considerably reduced, and its usefulness will not be decreased, if both impact and frequency estimates are rounded to the factors of ten shown in figure 1. There will be no significant difference in the overall exposure whether the damage from a certain event is estimated at $110,000 or $145,000. Assigning value to such things as loss of career caused by disclosure of confidential data or suffering caused by undue delay in the delivery of an annuity check is, in fact, more readily done in orders of magnitude than in actual figures. Here again, there will be no difference if the frequency of an event is expected to be twelve times a year or thirty. Using the scales for frequency from figure 1 will avoid the use of unwieldy fractions and maintain the flexibility to work with high probability events in days and low probability events in years.

IMPACT:

$10
$100
$1000
$10,000
$100,000
$1,000,000
$10,000,000
$100,000,000

FREQUENCY:

Once in 300 years
Once in 30 years
Once in 3 years (1000 days)
Once in 100 days
Once in 10 days
Once per day
10 times per day
100 times per day

FIGURE 1. *Orders of Magnitude of Estimated Impact and Frequency.*

4.3 Annual Loss Exposure

If the impact of an event, i.e., the precise amount of damage it could cause, and the frequency of occurrence of that event, i.e., the exact number of times it could happen, could be specified, the product of the two would be a statement of loss, or

Loss = Impact × Frequency of Occurrence.

However, because the exact impact and frequency can usually not be specified, it is only possible to approximate the loss with an annual loss exposure (ALE), which is the product of estimated impact in dollars (I) and estimated frequency of occurrence per year (F).

For ease in use, the orders of magnitude for estimated impact and estimated frequency of occurrence can be indexed, as shown in figure 2.

When i and f are indices to possible orders of impact and frequency,

the relationship of i to I is $I = 10^{i}$ and

the relationship of f to F is $F = \frac{10^{(f-3)}}{3}$

or $$F = \frac{10^{f}}{3000}.$$

Thus the annual loss expectancy can be calculated by the formula

$$ALE = \frac{10^{i}}{3} \times 10^{(f-3)},$$

which reduces to

$$ALE = \frac{10^{(f+i-3)}}{3}.$$

Using the table shown in figure 3 will be faster than following the formula for ALE but will produce the same result. Find the appropriate row and column for the i and f selected from figure 2; the cell where they intersect will contain the ALE.

If the estimated cost impact of the event is

$10, let i = 1
$100, let i = 2
$1000, let i = 3
$10,000, let i = 4
$100,000, let i = 5
$1,000,000, let i = 6
$10,000,000, let i = 7
$100,000,000, let i = 8

If the estimated frequency of occurrence is

Once in 300 years, let f = 1
Once in 30 years, let f = 2
Once in 3 years, let f = 3
Once in 100 days, let f = 4
Once in 10 days, let f = 5
Once per day, let f = 6
10 times per day, let f = 7
100 times per day, let f = 8

FIGURE 2. *Tables for Selecting of Values of i and f.*

Values of F

Values of i	1	2	3	4	5	6	7	8
1					$300	$ 3k	$ 30k	$300k
2				$300	3k	30k	300k	3M
3			$300	3k	30k	300k	3M	30M
4		$300	3k	30k	300k	3M	30M	300M
5	$300	3k	30k	300k	3M	30M	300M	
6	3k	30k	300k	3M	30M	300M		
7	30k	300k	3M	30M	300M			

Values of ALE

FIGURE 3. *Table for Determining Values of ALE.*

The tables from figures 2 and 3 can be combined as shown in figure 4 for greater convenience.

		Once in 300 yrs (100,000 days)	Once in 30 yrs (10,000 days)	Once in 3yrs (1,000 days)	Once in 100 days	Once in 10 days	Once per day	10 per day	100 per day
	f= i=	1	2	3	4	5	6	7	8
$10	1					$300	$3,000		$300k
$100	2				$300	$3,000	$30k	$300k	$3M
$1000	3			$300	$3,000	$30k	$300k	$3M	$30M
$10,000	4		$300	$3,000	$30k	$300k	$3M	$30M	
$100,000	5	$300	$3,000	$30k	$300k	$3M	$30M	$300M	
$1,000,000	6	$3,000	$30k	$300k	$3M	$30M	$300M		
$10,000,000	7	$30k	$300k	$3M	$30M	$300M			
$100,000,000	8	$300k	$3M	$30M	$300M				

FIGURE 4. *Combined Matrix of i, f and ALE.*

4.4 Procedure

The team will need an organized way of approaching their task and an orderly method of recording their findings. It would probably be impossible for the team to conceive of every event which could have a deleterious effect on data processing. Therefore, the risk analysis task is better approached from the standpoint of the data files, or applications systems, of which there is a finite number. Cataloging each data file or application system on a worksheet on which the results of the analysis can also be noted will give structure to the task.

Worksheets may be formatted in any way an agency finds useful, but they should be as simple as possible and should not contain any superfluous data. The worksheet shown in figure 5 can be copied and enlarged to provide more working space, if desired.

All of the organization's application systems, or data files arranged by application, should be listed on the worksheet(s). By tracing the flow of data through a system, the team will be able to pinpoint where in the processing the threats identified in the preliminary study could occur. Because of the preliminary vulnerability study and the team's collective familiarity with the systems applications files, they should be able to assign reasonable estimated frequencies to such events. If a file is used with more than one application system, it should be listed under each, as it can be vulnerable to different hazards under different systems.

Organizations with a large number of files will probably want in their initial risk analysis to consider their data on an application basis rather than on a file basis because of the size of the task awaiting them. Such an analysis should be followed by the more detailed file-by-file consideration in any instances where there is an indication that protection requirements differ radically among the files in any one application system.

The values of i and f should be filled in at each intersection on the worksheet, as should the value of ALE, or it will be impossible to reconstruct the basis for a particular ALE. Keep a running total of the ALEs attributable to each threat on the list of actual threats. (If additional threats surface, they should be added to the list.) A note in the "Comments" column linking the ALE to the particular threat, or threats, will be useful at the time of selecting remedial measures.

The effect of currently installed protective measures on undesirable events should not be taken into account at this stage. Their consideration would require efficacy judgments which are properly a part of the subsequent safeguard selection process.

Where more than one circumstance can affect data integrity, data confidentiality or processing availability, the i and f values for these events should be noted separately; this will be an aid in deciding on security measures. Use the "Comments" column to note the steps or functions in a system where problems can occur. When the team is considering data confidentiality, their task can be simplified by first eliminating the files which are known to contain no personal, proprietary or other information of a nature which would make disclosure a problem.

The further division of data integrity into modification and destruction is necessary because the two will not always have the same impact, nor occur with the same frequency.

The "Comments" column can be used as shown in figures 6 and 7 to indicate the processing step in which a destructive event can occur. It can also be used to refer to additional notes which may be needed to explain certain situations more completely.

The time periods in the "Processing Availability" column are mission dependent and will have to be determined by each organization for itself. They will be important in the selection of backup facilities and should be subject to review by top management. The destruction of equipment should be considered under "Processing Availability" because the ultimate effect of destroying equipment will be the inability to process data. The impact will be the cost associated with the inability to process rather than the cost of replacing equipment. Replacement is a possible remedial measure, the cost of which should be subjected to the same analysis as any other measure (as described in Chapter 6).

4.5 Special Advice

ADP risk analysis is a technique which relies heavily on the intuition, experience and technical knowledge of the team members. The

SYSTEM/APPLICATION Data Files	DATA INTEGRITY		DATA CONFIDENTIALITY	PROCESSING AVAILABILITY			COMMENTS
	Modification	Destruction					
	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	

FIGURE 5. *Risk Analysis Worksheet.*

452

comments in this section are included for the purpose of putting certain problems in perspective and giving the team confidence in its own collective judgment in areas where there often appears to be little or no precedent or guidance on which to base a decision.

4.5.1 Human Frailty

The team will come upon doubts as they weigh the part personal integrity plays in the security of a system. While every Federal employee who works in an ADP environment must have a clearance appropriate to the content or purpose of the systems he deals with, there is no way of knowing at any time what stresses are operating on an individual—what pressures he has at home, what jealousies exist in the work situation, what financial burdens he is under. For these reasons, it is usually best to leave individual personal integrity out as a factor contributing to security in a risk analysis. The right time for considering personal integrity is during development of the security plan, when various safeguards can be discussed.

Several general conclusions seem to be emerging from the growing body of statistics on computer crime [10]:

- The vast majority of white collar crime is committed by employees defrauding their own employers.
- In general, employees who defraud their employers do so using resources to which they have access in the course of their jobs.
- The best deterrent to white collar crime has proved to be curtailment of incentive, i.e., limiting the profit potential of dishonest activity to the minimum consistent with the assigned task. If employees can expect no more than minimal gain from unscrupulous acts, they will be less likely to attempt them. The second-best deterrent is the fear of getting caught. If employees know there is adequate surveillance of activity, they will be less apt to place themselves in jeopardy.

4.5.2 Physical Security/Inability to Process

Another difficulty the team can encounter is the confusion caused by treating fires, floods and other natural disasters solely as physical security problems. While the initial impact of natural disasters usually is physical destruction, there can be other less immediately obvious effects on processing capability, such as loss of utilities, loss of the services of key personnel and damage to data storage media. There can also be loss of services without any damage to a facility.

The loss of the physical facility and the loss of processing availability should be treated independently of each other, since neither necessarily causes the other. The total inability to process can be caused by circumstances other than physical destruction. For instance, hardware malfunctions can hold up all processing for several days; accidental erasure of critical programs or data can delay an urgent task for many hours; a fire in another building can deprive the ADP center of utilities; waterlogging of preprinted output forms can halt output until the forms can be replaced, possibly a matter of weeks. Flood damage can result not only from overflowing rivers, but also from leaky fixtures, bursting pipes or fire fighting activity nearby.

4.5.3. Estimating Frequency of Occurrence

At first the team may feel that estimating frequency of events for which there is no history of occurrence is impossible. Common sense, however, can help. Consider, for example, a payment system with good automated controls over the number of checks and the sums of the amounts of the checks. Between a hundred and a thousand people may know that it is relatively easy to change a recipient's address without risk that it will be detected; one of them could easily divert checks to an address where they could be picked up and cashed by someone other than the intended recipient. Such a situation should yield an estimated frequency much higher than once in thirty years and probably much lower than once every ten days, leaving the choice between once every three years and once every hundred days. Selecting the most appropriate of these figures depends on several factors, including the general atmosphere in which the system functions. If the number of people who know of the vulnerability is one or two hundred, the former is the most likely figure. If the number of people who know is nearer a thousand, or if employee dishonesty is accepted by management as long as it stays within established bounds, then the higher estimated frequency would be more likely.

4.6 Sensitivity of Documentation

All reports, worksheets and any other documentation or notes dealing with the risk analysis should be treated as highly sensitive and should be so marked by whatever method the organization uses.

5. AN EXAMPLE

An example using a hypothetical government agency has been developed to show some of the facets which must be considered in a risk analysis. Only a small part of the agency's total ADP applications are considered here.

5.1. General Environment

5.1.1. Central Computer Facility

- The central ADP facility is housed in a separate three-story wing of the agency's headquarters in central Kansas.
- The equipment consists of a large-scale processor with 3 CPUs, 32 tape drives, 10 billion characters of disk storage, 3 front-end communications processors capable of handling a total of 175 terminals (125 are presently in the system), a COM unit and a library of 50,000 reels of tape. Transmission between central facility and terminals is by private leased line.
- Guards check all personnel into and out of the computer area. Badges are required. Areas not monitored by guards are controlled by an electronic card system. Procedures are in effect covering lost, forgotten, stolen and damaged badges and card passes and the issuance of badges to visitors.
- There is a supervised fire detection/suppression system consisting of products-of-combustion detectors and a dry-pipe sprinkler system. Hand extinguishers are located throughout the facility, the type determined by the equipment or supplies in their vicinity. Continuing emergency team training is required of all computer operations personnel. The training includes actual use of the various extinguishers. Fire safety orientation is given to all employees when first hired and annually thereafter. Areas of the building adjacent to the computer facility do not have fire detection devices. These areas are under the control of operating units other than data processing.
- There is no emergency power or uninterruptible power supply backup. In the last seven years, the facility has experienced machine failure due to power outages resulting from thunderstorms, a fire at the utility substation and breaks in the main power feeder caused by a construction project. In recent months (especially summer) local brownouts have caused the failure of certain electronic equipment. These brownouts occur about every three weeks.
- The air conditioning unit is five years old and has suffered three breakdowns: one 2 years after installation, one 18 months later, and a third after another year. Two 100-ton cooling towers are located on the roof of the wing in which the ADP facility is located.
- Plastic covers are supplied for all hardware in the facility. The flooring is raised 24 inches and there are automatic pumps in case of water entry. The tape library is well protected from water damage.
- Emergency power-down switches are provided for all computer and air conditioning systems.
- Management is aware that, annually, about 400 tape and/or disk files are misplaced or destroyed by improper handling or overwriting because of incorrect labeling.
- Employee morale is notably high. The agency has established good personnel policies and the procedures for dealing with

employee complaints work fairly and to the satisfaction of most. All ADP personnel are aware of management's continuing interest in maintaining and enforcing security procedures at both central and remote facilities.

- The operating system must be restarted several times a week. Sometimes the problem can be traced to a hardware failure, but usually it is not resolved. Systems programmers maintain the system with little direct supervision. There is no formal review before changes are installed. The operators have learned how to keep the system running efficiently, but some of the evening and night supervisors have little understanding of what the operators do.

5.1.2 Terminals

- The remote job entry terminals are all located in GSA leased spaces, one at each field office. They are locked when unattended; however, they are used by several branches of the agency for a number of systems. Magnetic tapes are secured in locked cabinets located in terminal rooms. Data tapes are retained for one month only. Source documents on microfilm are stored in secure areas other than in the terminal room. Data are not protected during transmission from terminals to central facility.

5.1.3 Backup Facilities

- No plans have been made for emergency backup of automatic data processing.

5.2 Specific Systems

5.2.1 Application 100

This application supports a mission stemming from an Executive Order requiring a report to be produced and published on the third Thursday of each month. It has been automated for ten years. A master file containing the most recent report must be updated monthly with new data transmitted from 30 field offices to the central facility. When the new data are merged, a new report is produced and distributed through controlled official channels.

The following set of circumstances is assumed for this application:

- The data are necessary to the Federal community. Their output can have an economic impact on the private sector if released early.
- At the field offices, the source documents are microfilmed after data have been translated into machine readable format (magnetic tape). Seven of the offices have their own microfilming equipment; twenty-three have it done on contract.
- Data transmission to central facility is accomplished during third shift operation (0001 to 0800) every Tuesday.
- If communications network is down, data tapes are flown to the central facility. Communications failure occurs an average of three times a year.
- Only ADP personnel with appropriate clearances are authorized to handle the data throughout the entire process.
- To date, there have been no known incidents of unauthorized access to or early release of the data.
- Copies of updated reports are stored at the central facility in a special locked cabinet and backup copies are stored at a GSA Records Center. The backup copies are maintained for three (3) cycles—current, plus two most recent months.
- A part of the final report, Section A, is created from some preliminary data. It must be available two days before the final data are transmitted so that analysis can be started. Updating the previous month's report requires preparation of the master tape. Certain other tape files must be used in this process; these include personnel assignment data, regional projects data and budget status data.
- Each of the elements is considered critical to the final product. At the conclusion of each stage, checks are made for errors which might have been introduced. No major errors have ever been detected. Errors which have been found are restricted primarily to the new data tapes created by field offices.
- If the system were to be violated, or if the report were to be late, some adverse impact would be felt in the stock markets. There would be embarrassment to the Govern-

ment on both national and international scenes.

- The data are of such importance to "outside" individuals that relatively senior personnel could be tempted to obtain prerelease information or cause the final report to miss the established publication date.
- All personnel involved are continuously observed by their managers for any signs of attitude change, deterioration in performance, or other indications of situations that could result in breaches to the security of this project.
- All corrections, updates, or modifications to the software systems are closely monitored and tested before final approval and subsequent incorporation into the master system.

The system consists of the six stages shown below:

INPUT	PROCESS	OUTPUT
Stage 1—Data preparation Source data	key to tape verify duplicate microfilm destroy source documents	Source data tape + 1 copy microfilm
Stage 2—Data transmission Source data tape	transmit verify	Change data tape
Stage 3—File maintenance Master tape (current) Change data tape	update verify duplicate tape	Master tape (new + 1 cy)
Stage 4—Section A creation Master tape Personnel assignment data Regional project data Budget status data	calculate format	Sect A report
Stage 5—Final report creation Master tape Personnel assignment data Regional projects data Budget status data	same as St. 4	Final report
Stage 6—Querying Master tape Personnel assignment data Regional projects data	search read	video display

The worksheet for this application is shown in figure 6.

SYSTEM/APPLICATION Data Files	DATA INTEGRITY						DATA CONFIDENTIALITY			PROCESSING AVAILABILITY									COMMENTS
	Modification			Destruction						2 hrs			24 hrs			72 hrs			
APPLICATION 100	(i)	(ALE)	(f)	(i)	(ALE)	(f)	(i)	(ALE)	(f)	(i)	(f)	(ALE)	(i)	(f)	(ALE)	(i)	(f)	(ALE)	
Source Documents	4		1	4		1	5	$3k	2[a]			–			–			–	a. key to tape, st. 1, Thr. 9
Source data tape	4	$30k	4[a]	3		2	5	$30k	3[a]			–			–			–	a. key to tape, st. 1, Thr. 9
Microfilm		–			–		5	$30k	3[a]			–			–			–	a. microfilm, st. 1 Thr. 9
Change data tape	3	$300	3[a]	3	$300	3[a]	5	$300k	4[a]			–	6	3[a]	$300k			–	a. transmit, st. 2, Thr. 21
Master tape	5	$30k	3	5	$30k	3[a,b]	7	$30M	4[a,b,c]	7	4[a,b,d]	$30M			–			–	a. entire operation, st. 4, Thr. 14 b. entire operation, st. 5, Thr. 14 c. read, st. 6, Thr. 14 d. power interrupt, op. shut down
Personnel assignment data	5	$300	1	5	$300	1	5	$300	1	5	3[a]	$30k			–			–	a. calculate, st. 4, 5, Thr. 19, 23
Regional projects data	8	$300	2[a]	8		1	4	$300	2[a]	5	3[a]	$30k			–			–	a. calculate, st. 4, 5, Thr. 19, 23
Budget status data	3		1	3		1	4		1	5	3[a]	$30k			–			–	a. calculate, st. 4, 5, Thr. 19, 23

FIGURE 6. *Risk Analysis Worksheet for Application 100.*

5.2.2 Application 870

This system is used to maintain and control the agency's plans and gross budgetary information for the most recent five years, the current year and the next five: ProgHist, CurrProg and AgPlans. The software consists of an agency developed program, PFiles, and a commercial proprietary program, WWWMod, which does the modeling required to choose the optimum course for future plans.

There are six video graphics terminals equipped with hard copy printers located in the offices of top management and a small control center with a large video screen in the office of the head of the agency to be used for displaying the results of on-line modeling at staff meetings. All files are mounted on-line during normal working hours. They are updated after every working day at 1:00 a.m. with the previous day's transactions—an average of ten, except during February and August when processing time jumps from 1.8 hours a month to 4 hours a month.

The system consists of the three stages shown below:

INPUT	PROCESS	OUTPUT
Stage 1—Daily file maintenance	(1 am, 1.8 hrs/mo, except Feb and Aug 4 hrs/mo)	
CurrProg	update files	CurrProg
PFiles	verify	
	duplicate	
Stage 2—Querying & modeling	(8 am to 5 pm daily)	
AgPlans	search files	video display
CurrProg	read files	printout
WWWMod	calculate	
Stage 3—Semiannual report creation	(during working hours, Feb and Aug)	
AgPlans	calculate	MBOfuture rpt
CurrProg	verify	(2cc only)
ProgHist	format	AgPlans
WWWMod	update AgPlans	ProgHist
PFiles	update ProgHist	video display
	verify	

The worksheet for this application is shown in figure 7.

SYSTEM/APPLICATION Data Files	DATA INTEGRITY		DATA CONFIDENTIALITY	PROCESSING AVAILABILITY			COMMENTS
	Modification	Destruction		2 hrs	24 hrs	72 hrs	
APPLICATION 870	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	(i) (f) (ALE)	
AgPlans	6 3 $300k	4 2 $300	7 4 $30M	–	–	–	
CurrProg	5 3 $30k	4 2 $300	6 3 $300k	–	–	–	
ProgHist	4 2 $300	–	–	–	–	–	
WWWMod	4 3 $3k	4 2 $300	4 3 $3k	–	–	–	
PFiles	4 2 $300	4 2 $300	4 2 $300	–	–	–	

FIGURE 7. *Risk Analysis Worksheet for Application 870.*

6. SELECTION OF SAFEGUARDS

6.1 Alternative Measures

In the process of deciding which protective measures will provide the best overall security, management should look first to procedural and physical safeguards. Procedural controls, especially when used in combination with physical barriers, produce the highest degree of security for the lowest cost of all forms of protection. They satisfy the requirements of the Privacy Act of 1974 as well as many other demands either dictated by prudence or mandated by regulations. Procedural measures range from screening of all applicants before employment to off-site storage of backup data to standards for program development to preparation and testing of contingency plans. Procedural measures are essential for filling the gaps between manual and automated processing, between human beings and systems hardware and software. They are very effective against accidents resulting from human negligence and against amateur thievery. They promote an atmosphere of managerial concern for data and processing security that tends to discourage all but the most determined felons.

Most measures are effective against more than one threat. Maintaining facility access logs is a method of controlling who goes into a facility, of knowing who is in a facility at a given time, and of preventing unauthorized removal of material from a facility. Encryption protects data both during transmission and while in storage. Audit trails furnish information for backup and recovery and also provide a basis for variance detection.

Returning to the examples in section 5: it was found in Application 870 that though the files were very seldom used, they were on line throughout working hours, which greatly increased their vulnerability. The protection needs could be greatly reduced if the files were only available to an application running on a dedicated processor. This could be handled on a scheduled basis or on a given amount of notice. A several million dollar exposure could be circumvented in this way for only the cost of reviewing the system requirements. In Application 100 the losses that could occur at the field offices were found to be minor and appeared upon examination to be of a nature which could be averted by implementing procedural measures. The largest losses that could occur in the system were related directly to the data on the master tape and to the availability of processing to convert the data on the master tape into the required report. It was obvious that safeguards for protecting these two areas would also have an advantageous effect on many of the smaller concerns that were noted.

System security measures should be contemplated only after it has been established that physical and procedural safeguards are insufficient to meet the organization's protective requirements. If an organization's needs dictate the use of software or hardware protection for some systems, then those measures can also be incorporated in the protection plan for systems with lesser requirements, provided the operating costs of those systems are not thereby inordinately increased.

6.2 ALE Reduction vs Cost

The cost of each measure should be considered in three different ways: first, vis-à-vis the ALE reduction it brings about, then the total cost of the combined measures should be considered in relation to the net ALE reduction, and finally the additional ALE reduction by each measure should be compared to its share of the total cost. The selection of security measures is also discussed in FIPS PUB 31[6], FIPS PUB 41[1] and NBS Special Publication 500-33[2].

By constructing a matrix such as shown in figure 8, the threats and the protective measures which could affect one or more of them can be displayed. The threats should be arranged in order of ALEs attributable to them (highest to lowest). Each intersection in the matrix should contain three pieces of information: (a) the estimated ALE reduction, (b) the annual cost of the measure, and (c) the resultant annual saving. Great precision is not necessary in arriving at these three figures. The annual cost of a measure is listed opposite the most serious threat which it affects; opposite any other threat which is affected only the increase in cost to cover that threat is noted.

THREATS	MEASURES	A	B	C	D
1	(a)	*$20,000	$20,000	$18,000	
	(b)	9,000	15,000	8,000	
	(c)	11,000	5,000	10,000	
2	(a)	10,000	12,000	—	
	(b)	—	—		
	(c)	10,000	12,000		
3	(a)	4,000	6,000	—	
	(b)	1,000	—		
	(c)	3,000	6,000		
4	(a)	2,000	4,000	—	
	(b)	5,000	2,000		
	(c)	−3,000	2,000		

---

*See paragraph above.

FIGURE 8. *Array of Remedial Measures vs Threats.*

Now is the correct time to evaluate the usefulness of existing security measures to the overall security of the facility. They should all be included in the matrix but only the annual maintenance costs need be considered, not the initial installation costs since those have already been expended. The cost of those which are not solely for the purpose of computer security should be prorated if possible. It is also the correct time to consider replacement costs. Replacement of equipment should be treated the same as any other remedial measure. It may develop that the cost of replacing equipment is less, in some cases, than protecting it.

Comparing all the measures which remedy the same threat (or lesser included threats) will show which one is the most cost effective in the given circumstances. In the matrix above, protective measure A, costing $10,000, provides a $24,000 saving against threats 1, 2 and 3 while measure B, costing $17,000, provides a $25,000 summed saving against threats 1, 2, 3 and 4. The two measures together, at a cost of $27,000, provide an ALE reduction of $31,000. However, the final comparison reveals that the $10,000 expenditure for measure A only produces an additional saving of $6,000 over that obtained by the $17,000 expenditure. In some circumstances it may be determined that the additional reduction is necessary; in other less sensitive situations, the cost saving will be adopted instead. In addition, care should be taken to insure that the measures chosen to counter certain threats do not increase the estimated frequency of other threats.

With all of the ALE reduction and cost figures arrayed, various combinations of safeguards can be considered tentatively until a satisfactory aggregation of security measures is achieved. The matrix will be useful in explaining to management why particular safeguards should be selected.

The matrix and all other material associated with the risk analysis should be treated as highly sensitive. Copies of all letters, papers, worksheets and reports prepared by the risk analysis team should be preserved for the information of those performing subsequent risk analyses.

APPENDIX

A. APPLICATION SYSTEM VULNERABILITIES

It will be useful to the team, as they consider applications systems and data files, to be aware of the many undesirable events which can have serious consequences. A number of situations to which applications systems are vulnerable are listed here, grouped according to common system organizational structures. The list is not intended to be all-inclusive but only to suggest the various kinds of vulnerabilities that may exist in each system.

1. ERRONEOUS OR FALSIFIED DATA INPUT. Erroneous or falsified input data is the simplest and most common cause of undesirable performance by an applications system. Vulnerabilities occur wherever data is collected, manually processed, or prepared for entry to the computer.

- Unreasonable or inconsistent source data values may not be detected.
- Keying errors during transcription may not be detected.
- Incomplete or poorly formatted data records may be accepted and treated as if they were complete records.
- Records in one format may be interpreted according to a different format.
- An employee may fraudulently add, delete, or modify data (e.g., payment vouchers, claims) to obtain benefits (e.g., checks, negotiable coupons) for himself. .
- Lack of document counts and other controls over source data or input transactions may allow some of the data or transactions to be lost without detection—or allow extra records to be added.
- Records about the data-entry personnel (e.g., a record of a personnel action) may be modified during data entry.
- Data which arrives at the last minute (or under some other special or emergency condition) may not be verified prior to processing.
- Records in which errors have been detected may be corrected without verification of the full record.

2. MISUSE BY AUTHORIZED END USERS. End users are the people who are served by the ADP system. The system is designed for their use, but they can also misuse it for undesirable purposes. It is often very difficult to determine whether their use of the system is in accordance with the legitimate performance of their job.

- An employee may convert Government information to an unauthorized use; for example, he may sell privileged data about an individual to a prospective employer, credit agency, insurance company, or competitor; or he may use Government statistics for stock market transactions before their public release.
- A user whose job requires access to individual records in a file may manage to compile a complete listing of the file and then make unauthorized use of it (e.g., sell a listing of employees' home addresses as a mailing list).
- Unauthorized altering of information may be accomplished for an unauthorized end user (e.g., altering of personnel records).
- An authorized user may use the system for personal benefit (e.g., theft of services).
- A supervisor may manage to approve and enter a fraudulent transaction.
- A disgruntled or terminated employee may destroy or modify records—possibly in such a way that backup records are also corrupted and useless.
- An authorized user may accept a bribe to modify or obtain information.

3. UNCONTROLLED SYSTEM ACCESS. Organizations expose themselves to unnecessary risk if they fail to establish controls over who can enter the ADP area, who can use the ADP system, and who can access the information contained in the system.

- Data or programs may be stolen from the computer room or other storage areas.
- ADP facilities may be destroyed or damaged by either intruders or employees.
- Individuals may not be adequately identified before they are allowed to enter ADP area.
- Remote terminals may not be adequately protected from use by unauthorized persons.
- An unauthorized user may gain access to the system via a dial-in line and an authorized user's password.
- Passwords may be inadvertently revealed to unauthorized individuals. A user may write his password in some convenient place, or the password may be obtained from card decks, discarded printouts, or by observing the user as he types it.
- A user may leave a logged-in terminal unattended, allowing an unauthorized person to use it.
- A terminated employee may retain access to ADP system because his name and pass-

word are not immediately deleted from authorization tables and control lists.

- An unauthorized individual may gain access to the system for his own purposes (e.g., theft of computer services or data or programs, modification of data, alteration of programs, sabotage, denial of services).
- Repeated attempts by the same user or terminal to gain unauthorized access to the system or to a file may go undetected.

4. INEFFECTIVE SECURITY PRACTICES FOR THE APPLICATION. Inadequate manual checks and controls to insure correct processing by the ADP system or negligence by those responsible for carrying out these checks result in many vulnerabilities.

- Poorly defined criteria for authorized access may result in employees not knowing what information they, or others, are permitted to access.
- The person responsible for security may fail to restrict user access to only those processes and data which are needed to accomplish assigned tasks.
- Large funds disbursements, unusual price changes, and unanticipated inventory usage may not be reviewed for correctness.
- Repeated payments to the same party may go unnoticed because there is no review.
- Sensitive data may be carelessly handled by the application staff, by the mail service, or by other personnel within the organization.
- Post-processing reports analyzing system operations may not be reviewed to detect security violations.
- Inadvertent modification or destruction of files may occur when trainees are allowed to work on live data.
- Appropriate action may not be pursued when a security variance is reported to the system security officer or to the perpetrating individual's supervisor; in fact, procedures covering such occurrences may not exist.

5. PROCEDURAL ERRORS WITHIN THE ADP FACILITY. Both errors and intentional acts committed by the ADP operations staff may result in improper operational procedures, lapsed controls, and losses in storage media and output.

Procedures and Controls:

- Files may be destroyed during data base reorganization or during release of disk space.
- Operators may ignore operational procedures; for example, by allowing programmers to operate computer equipment.
- Job control language parameters may be erroneous.
- An installation manager may circumvent operational controls to obtain information.
- Careless or incorrect restarting after shutdown may cause the state of a transaction update to be unknown.
- An operator may enter erroneous information at CPU console (e.g., control switch in wrong position, terminal user allowed full system access, operator cancels wrong job from queue).
- Hardware maintenance may be performed while production data is on-line and the equipment undergoing maintenance is not isolated.
- An operator may perform unauthorized acts for personal gain (e.g., make extra copies of competitive bidding reports, print copies of unemployment checks, delete a record from journal file).
- Operations staff may sabotage the computer (e.g., drop pieces of metal into a terminal).
- The wrong version of a program may be executed.
- A program may be executed using wrong data or may be executed twice using the same transactions.
- An operator may bypass required safety controls (e.g., write rings for tape reels).
- Supervision of operations personnel may not be adequate during non-working hour shifts.
- Due to incorrectly learned procedures, an operator may alter or erase the master files.
- A console operator may override a label check without recording the action in the security log.

Storage Media Handling:

- Critical tape files may be mounted without being write protected.

- Inadvertently or intentionally mislabeled storage media are erased. In a case where they contain backup files, the erasure may not be noticed until it is needed.
- Internal labels on storage media may not be checked for correctness.
- Files with missing or mislabeled expiration dates may be erased.
- Incorrect processing of data or erroneous updating of files may occur when card decks have been dropped, partial input decks are used, write rings mistakenly are placed in tapes, paper tape is incorrectly mounted, or wrong tape is mounted.
- Scratch tapes used for jobs processing sensitive data may not be adequately erased after use.
- Temporary files written during a job step for use in subsequent steps may be erroneously released or modified through inadequate protection of the files or because of an abnormal termination.
- Storage media containing sensitive information may not get adequate protection because operations staff is not advised of the nature of the information content.
- Tape management procedures may not adequately account for the current status of all tapes.
- Magnetic storage media that have contained very sensitive information may not be degaussed before being released.
- Output may be sent to the wrong individual or terminal.
- Improperly operating output or post-processing units (e.g., bursters, decollators or multipart forms) may result in loss of output.
- Surplus output material (e.g., duplicates of output data, used carbon paper) may not be disposed of properly.
- Tapes and programs that label output for distribution may be erroneous or not protected from tampering.

[illegible] **PROGRAM ERRORS.** Applications programs should be developed in an environment that requires and supports complete, correct, and consistent program design, good programming practices, adequate testing, review, and documentation, and proper maintenance procedures. Although programs developed in such an environment will still contain undetected errors, programs not developed in this manner will probably be rife with errors. Additionally, programmers can deliberately modify programs to produce undesirable side effects or they can misuse the programs they are in charge of.

- Records may be deleted from sensitive files without a guarantee that the deleted records can be reconstructed.
- Programmers may insert special provisions in programs that manipulate data concerning themselves (e.g., payroll programmer may alter his own payroll records).
- Data may not be stored separately from code with the result that program modifications are more difficult and must be made more frequently.
- Program changes may not be tested adequately before being used in a production run.
- Changes to a program may result in new errors because of unanticipated interactions between program modules.
- Program acceptance tests may fail to detect errors that only occur for unusual combinations of input (e.g., a program that is supposed to reject all except a specified range of values actually accepts an additional value).
- Programs, the contents of which should be safeguarded, may not be identified and protected.
- Code, test data with its associated output, and documentation for certified programs may not be filed and retained for reference.
- Documentation for vital programs may not be safeguarded.
- Programmers may fail to keep a change log, to maintain back copies, or to formalize recordkeeping activities.
- An employee may steal programs he is maintaining and use them for personal gain (e.g., sale to a commercial organization, hold another organization for extortion).
- Poor program design may result in a critical data value being initialized twice. An error may occur when the program is modified to change the data value—but only changes it in one place.
- Production data may be disclosed or

destroyed when it is used during testing.

- Errors may result when the programmer misunderstands requests for changes to the program.
- Errors may be introduced by a programmer who makes changes directly to machine code.
- Programs may contain routines not compatible with their intended purpose, which can disable or bypass security protection mechanisms. For example, a programmer who anticipates being fired inserts code into a program which will cause vital system files to be deleted as soon as his name no longer appears in the payroll file.
- Inadequate documentation or labeling may result in wrong version of program being modified.

7. OPERATING SYSTEM FLAWS. Design and implementation errors, system generation and maintenance problems, and deliberate penetrations resulting in modifications to the operating system can produce undesirable effects in the application system. Flaws in the operating system are often difficult to prevent and detect.

- User jobs may be permitted to read or write outside assigned storage area.
- Inconsistencies may be introduced into data because of simultaneous processing of the same file by two jobs.
- An operating system design or implementation error may allow a user to disable audit controls or to access all system information.
- The operating system may not protect a copy of information as thoroughly as it protects the original.
- Unauthorized modification to the operating system may allow a data entry clerk to enter programs and thus subvert the system.
- An operating system crash may expose valuable information such as password lists or authorization tables.
- Maintenance personnel may bypass security controls while performing maintenance work. At such times the system is vulnerable to errors or intentional acts of the maintenance personnel, or anyone else who might also be on the system and discover the opening (e.g., microcoded sections of the operating system may be tampered with or sensitive information from on-line files may be disclosed).
- An operating system may fail to record that multiple copies of output have been made from spooled storage devices.
- An operating system may fail to maintain an unbroken audit trail.
- When restarting after a system crash, the operating system may fail to ascertain that all terminal locations which were previously occupied are still occupied by the same individuals.
- A user may be able to get into monitor or supervisory mode.
- The operating system may fail to erase all scratch space assigned to a job after the normal or abnormal termination of the job.
- Files may be allowed to be read or written without having been opened.

8. COMMUNICATIONS SYSTEM FAILURE. Information being routed from one location to another over communication lines is vulnerable to accidental failures and to intentional interception and modification by unauthorized parties.

Accidental Failures:

- Undetected communications errors may result in incorrect or modified data.
- Information may be accidentally misdirected to the wrong terminal.
- Communication nodes may leave unprotected fragments of messages in memory during unanticipated interruptions in processing.
- Communication protocol may fail to positively identify the transmitter or receiver of a message.

Intentional Acts:

- Communications lines may be monitored by unauthorized individuals.
- Data or programs may be stolen via telephone circuits from a remote job entry terminal.
- Programs in the network switching computers may be modified to compromise security.
- Data may be deliberately changed by individuals tapping the line (requires some

sophistication, but is applicable to financial data).

- An unauthorized user may "take over" a computer communication port as an authorized user disconnects from it. Many systems cannot detect the change. This is particularly true in much of the currently available communication equipment and in many communication protocols.
- If encryption is used, keys may be stolen.
- A terminal user may be "spoofed" into providing sensitive data.
- False messages may be inserted into the system.
- True messages may be deleted from the system.
- Messages may be recorded and replayed into the system ("Deposit $100" messages).

REFERENCES AND SUGGESTED READING

1. Computer Security Guidelines for Implementing the Privacy Act of 1974, National Bureau of Standards (U.S.), Federal Information Processing Standards Publication (FIPS PUB) 41, National Technical Information Service, Springfield, VA (1975).
2. Courtney, Robert H., Jr. and Orceyre, Michael J., Considerations in the Selection of Security Measures. National Bureau of Standards (U.S.) Special Publication 500-33, Government Printing Office, Washington, DC (1978).
3. Data Security Controls and Procedures—A Philosophy for DP Installations. G320-5649-00, IBM Corporation, White Plains, NY (1976).
4. Disaster Preparedness, Office of Emergency Preparedness Report to Congress, Stock Number 4102-0006, Government Printing Office, Washington, DC (1972).
5. Glossary for Computer Systems Security, National Bureau of Standards (U.S.), Federal Information Processing Standards Publication (FIPS PUB) 39, National Technical Information Service, Springfield, VA (1976).
6. Guidelines for ADP Physical Security and Risk Management, National Bureau of Standards (U.S.), Federal Information Processing Standards Publication (FIPS PUB) 31, National Technical Information Service, Springfield, VA (1974).
7. Hoyt, Douglas B., Ed., Computer Security Handbook, McMillan Information, New York (1973).
8. Martin, James, Security, Accuracy and Privacy in Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, NJ (1973).
9. Nielson, N. R., Ruder, B., Madden, J. D. and Wong, P. J., Computer System Integrity, SRI International, Menlo Park, CA (1978).
10. Parker, Donn B., Crime by Computer, Charles Scribner's Sons, New York (1976).
11. Wong, K. K., Computer Security Risk Analysis and Control, Hayden Book Company Inc., Rochelle Park, NJ (1977).

C Money

FIPS PUB 73

FEDERAL INFORMATION
PROCESSING STANDARDS PUBLICATION

1980 JUNE 30

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

Contingency Planning – Pg 42

GUIDELINES FOR SECURITY OF COMPUTER APPLICATIONS

CATEGORY: ADP OPERATIONS
SUBCATEGORY: COMPUTER SECURITY

1. INTRODUCTION

Computer security concerns the protection of computers and their services from all natural and human-made hazards. As computers perform more and more critical services and as effective human review becomes more difficult, the most serious security concerns often become a matter of performance assurance—an assurance that the computer performs its critical functions correctly and that there are no harmful side-effects [PATRR 78].

[redacted]RCBCC 79]. Previous guidelines [FIPSP 74] dealt extensively with security against physical threats to the computer facility[redacted]

The term "security" is sometimes used in a narrower sense to cover only protection against unauthorized disclosure of information[redacted]

These guidelines focus on controls for use with computer applications. Inadequacies in the design and operation of computer applications [BOEHB 73A], [FIFED 77], [MCCAJ 77A], [MCCAJ 77B], [MCCAJ 77C], [YOURE 75], [MILLH 77] are a very frequent source of harmful effects associated with computers, and in most cases the effort to improve security should concentrate on the application systems. An exception occurs when the nation's security or well-being could be seriously affected. In such situations, there is a significant danger that flaws inherent in the operating system could be exploited to avoid controls built into the application system; and for these applications security should be built in from the bottom up—beginning ~~with the~~ hardware and the operating system. Until operating system security improves to the point that operating systems can be trusted, the majority of Federal applications must coexist with current operating systems. ~~For these~~ applications, vast improvements in security are usually possible before inadequacies in the operating system become a limiting factor.

Key

Key

Security concerns should be an integral part of the entire planning, development, and operation of a computer application. Much of what needs to be done to improve security is not clearly separable from what is needed to improve the usefulness, reliability, effectiveness, and efficiency of the computer application. These guidelines are compatible with other FIPS guidance and with best current practices for achieving these other goals. On the other hand, many tradeoffs must be made between security and other goals. The tradeoffs need not always be resolved in favor of security, but security requirements must be appropriately weighed in each tradeoff, and they cannot be consistently neglected.

Security is given special treatment in these guidelines because it is a new and difficult problem to some users and managers and it is frequently neglected until it is too late to take the most effective action. Many of the hazards associated with computer applications do not occur frequently, and most managers have not had firsthand experience with them. It is easy for managers to neglect these hazards and become enmeshed in the smaller, more immediate, and more obvious problems.

Section 2 describes the multifaceted nature of computer security and identifies the differences in objectives, sensitivity levels, and vulnerabilities that must be considered. Section 3 is more technical and describes the basic controls that are commonly applicable. Readers who are applying these guidelines to a specific application system may skim this section and then later use it as reference material on the specific topics that are useful in their application. The remainder of the document—sections 4 to 7—describe the security-related decisions that should be made at each stage in the life cycle of a computer application. Section 4 introduces the life cycle approach to security—both for new systems and for revisions to existing systems. Section 5 identifies the questions about security feasibility and risk assessment that should be asked during the initial planning for an application system. Section 6 describes the decisions about security that should be made at each stage of the system development and it describes controls that should be enforced during the development process. Section 7 deals with the security provisions to be enforced during the day-to-day operation of the system.

2. SECURITY OBJECTIVES AND VULNERABILITIES

Computer security is needed because undesirable events during computer processing can have effects such as denial of benefits due to citizens, unauthorized disclosure of sensitive information, loss of Government money or resources, human injury or illness, false arrest, and, in extreme cases, loss of life and endangerment of the national welfare and security. Since the possible effects are so different, specific security objectives must be determined for each application.

This section describes general ways of classifying security objectives and illustrates some of the differences among various applications.

2.1 Integrity, Confidentiality, and Availability

To identify objectives that are common to many diverse computer applications, it is useful to classify undesirable events not in terms of their ultimate effects but in terms of their proximate effects on computerized data. An event may result in unauthorized or undesirable

- modification or destruction of data,
- disclosure of data,
- unavailability of data or system services.

This classification of undesirable events corresponds to the following three security objectives for computer systems:

1. DATA INTEGRITY—The state that exists when computerized data is the same as that in the source documents or has been correctly computed from source data and has not been exposed to accidental or malicious alteration or destruction. Erroneous source data and fictitious additions to the data are also considered violations of data integrity.
2. DATA CONFIDENTIALITY—The state that exists when data is held in confidence and is protected from unauthorized disclosure. Misuse of data—by those authorized to use it for limited purposes—is also a violation of data confidentiality.
3. ADP AVAILABILITY—The state that exists when required ADP services can be obtained within an acceptable period of time[1].

Many controls [RUTHZ 77], [RUTHZ 78], [EDPAF 77], [IIASA 77], [SGCCA 75B], [MAIRW 76] are effective in meeting all three security objectives; however, some are specific to one objective and may even have a detrimental effect on other objectives. For example, the damage resulting from loss or destruction of data (a data integrity problem) can be reduced by keeping multiple copies of the data—but this makes data confidentiality more difficult. Thus it is important to determine how much weight must be put on each of the three objectives of data integrity, data confidentiality, and ADP availability.

2.2 Accidental and Deliberate Events

A second classification of objectives arises from the differences between accidental and deliberate acts. Accidents, errors, and omissions generally account for more losses than deliberate acts; and thus they should be the first focus of attention. Safeguards that reduce the potential for harmful effects from accidents are also a first step toward reducing the opportunities for fraud and deliberate misuse—a system that tolerates frequent errors is a fertile ground for criminal activity that can be masked by the errors.

While controls that reduce accidental failings are the first step, many systems must also be protected against deliberate fraud and misuse [BEQUA 78A], [BEQUA 78B], [ALLEB 76], [PARKD 73], [PARKD 75], [PARKD 76], [RCBCG 76B], [KRAUL 79]. For some applications, existing legal sanctions[2] along with external efforts to identify criminal activity may be adequate; however, for a growing number of applications, it is necessary to include internal controls to prevent or at least detect criminal activity within the application.

Controls against deliberate acts must be carefully thought out because deliberate acts can exploit the weakest points in the control system. Controls that do not reduce the vulnerability of the weakest points will do little to deter deliberate acts. To protect a house against accidental rain damage, each window that is closed reduces the potential damage; to protect the house against a burglar, it does little good to lock second-story windows unless *all* the weaker points on the first story are protected.

Security against deliberate acts must control all the most likely avenues of attack; however, substantial deterence against fraud and embezzlement can be achieved if a potential perpetrator believes there is a definite, however small, probability of being detected and apprehended. Many potential white-collar criminals will be deterred when it is clear that the act is illegal and that there is no way to commit the crime without some distinct chance of being detected and prosecuted. Unfortunately, even this limited goal is difficult to achieve. Computers are very predictable and they follow their programmed instructions with great regularity; these properties are very useful to a criminal in preserving the appearance of normality while committing a crime.

---

[1] Equipment Security—the protection of the investment in the computer itself and in related equipment—is a fourth area of concern; however, this is usually separate from the security concerns of the application system [FIPSP 34], [RCBCG 76C].

[2] Current legal sanctions for computer-related crime are weak and are often difficult to enforce.

[redacted]

The degree of sensitivity of an application depends on the nature of the data it processes and on the way the application system is used to support other activities. Each agency must determine the degree to which its computer applications are sensitive. Examples of applications that might be sensitive are given below. The groupings reflect systems that have somewhat similar security problems; they do not necessarily reflect increasing levels of sensitivity.

2.3.1 Applications Providing General Processing Support

MAJOR OBJECTIVES: data integrity

EXAMPLES:

- Engineering calculations used in aircraft design
- Query systems that support health care decisionmaking
- Automated scheduling of safety inspections
- Simulation of the dispersion of toxic substances

In these applications, accidents are a major concern. Health, safety, welfare, and even lives may depend on the correctness of the output from these applications. Thus data integrity—including integrity of the programs—is critical; however, if the application routinely produces accurate and useful results, the probability of traumatic consequences from a random error may be low as long as gross errors are likely to be detected manually before they result in serious harm. The primary concern is for accidents, errors, and omissions. Deliberate acts could occur, but there is little motivation for them, and those responsible for the system may well decide that security efforts will be more effective if they are concentrated on oversights and errors. If personal or proprietary data is involved, then data confidentiality is a concern. ADP availability will be a requirement for some of these applications.

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

2.3.3 General-Purpose Information Systems

MAJOR OBJECTIVES: data integrity and data confidentiality.

EXAMPLES:

- Generalized data management systems
- Centralized management information systems

These applications, which serve large and varied user populations, have the data integrity concerns of any of the above systems. Furthermore, the simultaneous use by different user populations makes data confidentiality more critical, and the accumulation of large quantities of data in the same system may make its misuse more attractive. The generality of these systems and their associated security requirements also make it more difficult to design effective security controls. Availability objectives come from the user groups that are most dependent on the system for their day-to-day functions [FIPSP 80B].

2.3.4 Automated Decisionmaking Systems

MAJOR OBJECTIVES: rigorous data integrity.

EXAMPLES:

- Fully automated funds disbursement and accounting systems
- Automated inventory reordering
- Automated scheduling for maintenance

In these systems manual review becomes more difficult, so that errors made by the automated system are less likely to be detected before they lead to serious harm. The degree of data confidentiality and processing availability required depends upon the sensitivity of the application's data and upon the consequences of delayed processing respectively [RCBCG 76A].

2.3.5 Real-Time Control Systems

MAJOR OBJECTIVES: ADP availability and data integrity.

EXAMPLES:

- Air traffic control
- NASA mission control
- Rapid transit system control
- Automated production control

These systems have all the problems of automated decisionmaking systems plus a rigorous requirement for constant availability and very rapid response times. The security controls discussed in this document are needed in these systems, but additional controls including automated fault detection, backup, and recovery along with redundant hardware support are also needed to meet all their security objectives.

2.3.6 Systems Affecting National Security or Well-Being

MAJOR OBJECTIVES: data integrity, confidentiality, and ADP availability.

EXAMPLES:

- Military command and control
- Management of multilevel classified information
- Integrated electronic funds transfer
- Nuclear material control and accountability

These systems must be protected against hostile acts by unauthorized persons who have considerable resources to devote to planning ways to compromise them. The approach to security discussed in these guidelines is not effective against this kind of threat. More effective techniques for formal development and verification of controls in both the operating system and the application, which are emerging from the research community, are necessary for improved protection in this area [BERST 79], [DODKS 78], [FEIER 79], [LINDT 76C], [MCCAE 79], [NEUMP 77B], [POPEG 78A], [POPEG 79B], [SCHIW 75], [WOODJ 77], [ARMY 78], [JEFFS 76], [NCEFT 77], [WULFW 74B].

2.4 Vulnerabilities

Information resident on a computer system is vulnerable to a wide variety of hazards. The selection of controls must be based not only on security objectives but also on all the vulnerabilities of the application. Some familiarity with these vulnerabilities is required in order to understand the problems being addressed by these guidelines. A few general examples are given below. The reader is referred to the appendix of [FIPSP 79B] for more extensive examples.

- Input errors—Data may be altered, disclosed, lost, or misidentified during input preparation and incorrect, misinterpreted, inconsistent, or unreasonable data may be accepted as valid for processing.
- Open system access—There may be no control over who can use the computer application.

- Poorly defined criteria for authorized access—The personnel in direct charge of the application may not know who should have access to various items of information.
- Unaudited access to data—If individuals can access data knowing that there will be no record of their actions, then they will feel they cannot be held accountable for their acts.
- Unprotected information—Data may not be protected from unauthorized access. This applies to on-line files and also to off-line files such as magnetic tapes. The latter are sometimes accessible simply by requesting that they be mounted.
- Dial-in access—Unauthorized persons may obtain access to the system—especially if remote, dial-in access is allowed.
- [redacted]—Programs can contain many undetected errors, especially when they are written with poor programming practices or are not extensively tested. A program error may result in undesirable modification, disclosure, or destruction of sensitive information.
- Mistaken processing of data—Processing requests may update the wrong data; for example, if a tape is mounted at the wrong time.
- Operating system flaws—Design and implementation errors in operating systems may allow a user to gain control of the system. Once in control, the user can disable controls, erase audit trails, and access any information on the system.
- Subverting programs—Programs containing hidden subprograms that disable security controls can be submitted. These programs can copy information into secret or misidentified files for use at a later time, or perform unauthorized modifications of data.
- Spoofing—Actions can be taken to mislead system personnel or the system software into performing an operation that appears normal but actually results in unauthorized access.

3. BASIC CONTROLS

This section describes the basic controls that can be used to achieve security objectives. Selection of the controls that are applicable and necessary for a given application system depends both on its security objectives and on the environment in which the system operates. Guidance on how to make this selection in an orderly way will be given in sections 4 to 7.

Some controls discussed in this section may be implemented partially by the hardware, by the operating system, or by the facility management. Any applicable controls that are already available should be understood and used. Some controls that are often supported by an operating system are discussed in this section because the boundary between the operating system and the application system varies considerably and because it is ultimately the application system designer's responsibility to ensure that the combined effect of all controls is adequate.

Controls that are often expensive are identified as such in their descriptions; however, the actual cost will depend on many factors unique to each facility. Since many controls will also improve the manageability and the reliability of the application, much of their cost should be considered an integral part of the development and operation of an ADP application.

3.1 Data Validation

PROBLEMS ADDRESSED:

> Invalid data leads to erroneous outputs and is usually the proximate cause of harmful human, social, and economic effects.
>
> Invalid data can destroy the credibility of the system, demoralize those trying to use it, cause excessive system maintenance costs, and, in extreme cases, cause the system to be unavailable or unusable.

Data validation involves the examination of computerized data to determine if it is accurate, complete, consistent, unambiguous, and reasonable. This section gives an overview of the direct methods that can be used to evaluate data in order to identify likely errors. Since these direct evaluation methods are not able to find all errors, data integrity also depends on the correctness and integrity of all the activities by which the data is collected and processed. The controls of sections 3.2 to 3.6 as well as those in sections 6 and 7 must also be used in order to prevent subtle errors from being introduced into the data. Data validation is a very basic control, but it should only be expected to detect gross errors and it will not compensate for poor control over other aspects of the application system.

Data should be validated:

- during data collection and entry—prior to its use by the system; and
- continuously, as new data is generated or used during processing.

Errors should be detected and corrected as soon as possible in order to prevent the propagation of invalid data throughout the system and the potential contamination of the system data base.

Once invalid data has been identified, it must be corrected. The process of correcting errors is itself very prone to further errors, and data validation methods should be applied thoroughly during the correction activity.

Input verification, consisting of operational procedures to assure that source data is accurately transcribed to machine-readable form, is an important aspect of data validation. Input verification is discussed in greater detail in section 7.1 along with other manual information handling controls.

3.1.1 Consistency and Reasonableness Checks. Relationships between data elements that arise from the intended meaning of the data can be used to identify errors in the source data as well as errors in data preparation and processing. Often past records that are already in automated form can be used to predict probable values for current records. Any major deviation from expected values can then be flagged for further investigation. The appropriate checking to identify suspicious values must be defined by those who are familiar with the application and must be included in the application's design; it cannot be left up to the programmers.

3.1.2 Validation During Data Entry. Automated editing and validation should be used in both batch and on-line systems. In a batch processing system, validation routines may be run against the input data before it is processed. Alternately, validation can occur as each transaction is processed with transactions that contain errors being recorded on a file for correction at a later time. On-line systems can provide the data entry personnel with immediate validation information so detected errors can be corrected immediately. In either system, checks that must also use information from the data base may not be practical until just before processing.

Basic techniques for editing and validating data during data entry are listed below. More thorough discussions of techniques can be found in [MARTJ 73], [MAIRW 76], [JANCE 74], and [IIASA 77].

- Fields can be checked for:
  - legitimate characters (format checks),
  - proper sequences with respect to corresponding fields in preceding data records,
  - reasonableness,
  - consistency with other fields in the record,
  - values within specified ranges or limits,
  - valid items or codes,
  - the validity of a self-checking number contained in an extra digit(s) appended to the field, and
  - cryptographic check digits (i.e., data authentication information appended to a field which is a cryptographic function of all the data in the field).
- Records or transactions can be checked for:
  - completeness,
  - consistency of fields in the record with items in previous transactions or with other items in the data base, and
  - valid sequences of transactions.
- A group of records or transactions can be checked by maintaining:
  - control totals (e.g., batch, hash, amount, document, running),
  - record/transaction counts,
  - batch number checks, and
  - item/transaction balances.

3.1.3 Validation During Processing. Further checking of input data during processing and validation of data that is generated by the application system are essential for assuring data integrity. Many of these techniques are also used by auditors to audit data integrity. All of the methods described previously for validation of input data can be applied during processing. In addition, the following methods can also be used. More detailed discussions of these can be found in [MARTJ 73], [MAIRW 76], [JANCE 74], and [HASA 77] referenced above.

- Integrated Test Facility (ITF). The ITF allows the performance of the application system to be monitored by incorporating dummy master records into the data base. Once these records are in place, test transactions which are included with live data can be processed against the dummy records during the normal processing cycle. Processing results of the test transactions can then be compared to predetermined results.
- System Control Audit Review Files (SCARF). Specifically designed tests—placed within the application's code—are performed on the processed data during normal processing. Some criteria are selected to extract desired records (e.g., processing exceptions, predetermined samples) and to write them onto a file which can be reviewed at a later time.
- Tracing provides the ability to follow (trace) specifically marked or tagged transactions as they are processed by the application system. It requires additional application codes and an extra field in the transaction for the tag. During processing, a record or trail is generated for the marked transaction which can be analyzed to determine if the processing was correct.
- Sampling techniques where portions of the data are inspected to determine their validity.
- Run number controls which account for each run of an application system against a data base. The run numbers kept in the data base and in the application program must agree.
- Reconciliation of control files. Important elements of the data base are identified and a file of transactions is maintained with a running total. Periodically, the old balance (taken from the last reconciliation) is added to the running total and must equal the new balance (taken from the data base).
- Off-line file scanning of direct access files during periods of their inactivity (e.g., at the end of the day) can be used to balance files.
- Rounding error controls to prevent fractions of cents from being stolen.

3.1.4 Data Element Dictionary/Directory. A data element dictionary/directory (DED/D) is a software tool associated with data base management systems which can be used to validate data elements in a data base. (In addition to its data validation function, it can also contain security authorization specifications to control access to data elements, and it can aid the computer auditing function.)

The DED/D can contain validation and redundancy-checking criteria in the form of formatted descriptive information defining both legal formats for data elements and expected relationships between data elements in the data base. More detailed descriptions of DED/D can be found in [LEONB 77], [ICST 77], and [FIPSP 80A].

3.1.5 Management Considerations. Data validation can be costly, and it may not catch the most subtle errors. Nevertheless, data validation is basic to good system design; it is not just a security control. Consequently, data validation costs should be included in the cost-benefit studies of the total system.

3.2 User Identity Verification

PROBLEMS ADDRESSED:

Identity verification is necessary if access to the system's data and resources is to be controlled.

Users cannot be held accountable for their actions unless they are positively identified.

User identity verification usually involves a two-step process:

- Identification which occurs when the user provides an identifier—the name by which the user is known to the system. The user's identifier is unique, is unlikely to change, and need not be kept secret. It is used during processing for authorization controls, for variance detection, and for other purposes such as accounting and billing.
- Verification which occurs when the individual passes some further test which "proves" that the user is actually the person associated with the identifier. This is also called user authentication.

3.2.1 Requirements for Identity Verification. Requirements for identity verification depend on the nature of the interface to the application system. For batch applications the requirement may be simply for visual identification and manual logging when the job is submitted. For applications with on-line users, it is usually necessary to require automated identity verification. Highly privileged actions such as modification of the software and changes to authorization tables usually should be allowed only from a controlled area where redundant visual and automated identity verification can be required.

Users should be uniquely identified so that they can be held responsible for their subsequent activities—it is usually not enough just to verify that the user is one of the authorized users. The only cases where it may not be necessary to have distinct verification test for each individual occur when:

- All approved users of a given interface are allowed to perform any of a (possibly restricted) set of actions (i.e., no authorization controls from sec. 3.3 are needed).
- There is no need to keep records of who performed what actions (i.e., no journalling and variance detection controls from secs. 3.4 and 3.5 are needed).

The second condition is seldom fulfilled since it implies that any authorized user can misuse the system anonomously and with very little risk of detection or that recovery from system failure does not require knowledge of which users performed what prior actions. Nevertheless, in situations where employees have to share a terminal to make brief queries, repeated identity verification could be burdensome. If there are physical and visual controls to ensure that unauthorized users cannot get to one of the terminals and if a careful analysis shows that journalling and variance detection on the queries is not cost effective, then unique identification of each user may not be necessary.

3.2.2 Basic Techniques. There are three basic techniques by which an individual can "prove" identity. The individual can present for verification:

- Information KNOWN
  - password
  - numerical combinations
  - other information known only to user
- Objects POSSESSED
  - magnetically encoded card
  - key for a lock
- PHYSIOLOGICAL, BEHAVORIAL, or MORPHOLOGICAL attributes
  - signature
  - fingerprints
  - voiceprints
  - appearance and facial characteristics

Of all three basic techniques, passwords (something KNOWN) are the most commonly used automated method of verifying a user's identity. If password generation and password distribution are effectively administered and if passwords are adequately protected by users and by the system, then passwords can be used to verify identity with a degree of reliability that is adequate for many applications. A more thorough discussion of password techniques is contained in [WOODH 77] and [HOFFL 77].

Of the three techniques, the last two automated approaches are usually more expensive than passwords but may be somewhat more reliable. When necessary, they can be used in combination with passwords to obtain a degree of reliability commensurate with the risks of most applications. A detailed description and guide for evaluating automated identity verification techniques can be found in [FIPSP 77B], [HOFFL 77], and [CARRJ 77].

3.2.3 Management Considerations. None of the verification techniques mentioned above are completely reliable since:

- passwords can be lost, stolen, or revealed,
- cards and keys can be lost, stolen, or duplicated, and
- devices used to recognize physiological attributes will accept a small percentage of impostors and reject a small percentage of authorized users.

Although strictly enforced administrative controls and user cooperation can greatly enhance the reliability of verification techniques, high reliability can be costly and entail an inconvenience for users. Consequently, identity verification requirements should be tailored to the actual needs of each system interface.

3.3 Authorization

PROBLEMS ADDRESSED:

> If several people are to use an application that maintains diverse information, then data confidentiality and data integrity will not be preserved unless each user is allowed access to only that information which is needed.

Once a user's identity has been verified, the application may still need to check each request for service to determine whether it is a legitimate request by that user. Some users may be authorized to perform some functions but not others and to have access to some data but not to other data [SALTJ 75], [DENND 79B], [REEDS 74], [BERGJ 72], [CARRJ 77].

An elementary level of authorization can be enforced if the application has different interfaces for different purposes. For example, if there are different interfaces for entering transactions, for generating reports, for updating data, and for modifying programs, then each user can easily be authorized to use some but not all of these interfaces.

More complex authorization systems will usually be needed in order to satisfy each user's actual needs without allowing unneeded access to the application's data, programs, or other resources. In general, the principle of "least privilege" should govern. The less a person using the system is allowed to do (consistent with the work the user is required to do), the safer will be the application's other users, and the individual's own data and resources.

3.3.1 Authorization Schemes. Authorization schemes will have to be tailored to the needs of the organization being served by the application. There are a wide variety of possible authorization schemes described in the literature on computer security [MARTJ 73], [HOFFL 77], [SALTJ 75], [FABBR 74], [FERNE 75A], [FERNE 75B], [GRAHG 72], [HOFFL 73], [CHAMD 75], [STONM 74], [REEDS 74], and [LINDT 76C]. In general, an authorization scheme is the means whereby management can control:

- WHO will have
- what MODES OF ACCESS to
- which OBJECTS and resources.

WHO may include not only the human users of the application but also terminals and modules within the application. In many cases access to some sensitive objects should only be allowed from certain terminals. It is also possible for an authorization system to control the interactions between modules internal to the application software. This can greatly enhance the integrity, reliability, and maintainability of the application software [LINDT 76C]; however, this use of authorization is not yet widespread.

The different MODES OF ACCESS typically include read, write, and execute access to data and programs. The allowable modes of access can be usefully extended to provide independent control over access to each function supported by the application or by its internal modules.

The OBJECTS which are candidates for authorization control include:

- Data objects (files, records, libraries, segments, etc.),
- Executable objects (commands, programs, etc.),
- Devices (terminals, printers, etc.),
- Storage media (tape reels, disk packs, etc.),
- Transactions,
- Control data within the application, and
- Named groups of any of the above elements.

The ability to create named groups of system elements is an important way to simplify the administration of an authorization scheme.

In some cases, the authorization decision may depend not only on WHO is requesting what MODE OF ACCESS to which OBJECT, but also on other easily testable conditions. The time of the day, the day of the week, previously detected security variances, or other concurrent activity might be used to affect the authorization decision. In some cases, not only the name of the object but also a current value stored in the object may be used to make the authorization decision. This is called data-dependent access control. Clearly, the more factors that are used in arriving at an authorization decision, the more complex will be the required mechanism.

An authorization scheme consists of two parts—a set of authorization data or "rules" and a mechanism that uses the authorization data to decide whether a given access request should be allowed or denied. The authorization data allows the authorization system to accommodate to change whereas the authorization mechanism is built in and is usually very difficult to change. A careful study may be needed to select an authorization scheme which can accommodate all likely requirements over the lifetime of the system and yet is not so complex that it is unreliable and hard to manage.

The authorization mechanism generally should be invoked when one named element refers to another (e.g., a user request for service, a transaction input, a call for execution, an application system call on the operating system). The mechanism should be invoked at the point where the controlling process resolves symbolic references.

3.3.2 Delegation of Authority. An authorization system should distinguish the right to interact with an object from the right to authorize others to interact with it. In general, the three authority levels:

- ability to interact,
- ability to authorize interactions (delegation), and
- ability to appoint those who may authorize interactions,

should be separately controlled. None should, except at management option, imply either of the others. This is analogous to the distinctly different authorities involved in entering a bank vault, guarding the vault (deciding who may enter), and appointing the guards.

Policies for delegating authority should be given special attention when designing an automated authorization system. Delegation policies that appear to be simple can have surprising secondary effects in an automated system. For example, consider an individual with authority to access an object and to delegate access rights to others. If the user is about to have access authorities rescinded, the consequences must be well understood. One consequence (depending upon design) might be that all users whose authority originates from that individual will lose their authority when the primary user's authority is removed. This may be acceptable in some cases, and it may be catastrophic in others. On the other hand, if the authorities originating from the primary user's authority are undisturbed and untraceable, this too can be catastrophic. The authorization mechanism should enable management to discern the ultimate effects of such removals or modifications of authority. This requires that the authorization data, including back-chained or derivative authorities and all grouping relationships, be displayable.

The concept of data "ownership" is often used in security applications. The "creator" or "custodian" of a data file is usually considered to be the "owner." Other people who use the data are given the right to read it or to update it under controlled circumstances. Access may be granted temporarily or permanently (i.e., a copy may be made and the recipient becomes the "owner" of the copy). Security provisions must take into account the desirability of implementing the concept of "ownership."

3.3.3 Protecting the Authorization Data. The authorization data must accurately reflect management's intentions with respect to authorized activities. The authorization data is the basis for controlling system activity and has great potential for disruption.

It should be simple for management to grant, modify, or rescind an authorization; and management should be able to display all the current authorizations. If these actions are complex or unwieldy, errors will be more likely, disruptions will be more frequent, and use of the authorization mechanism will be less likely. In the past, simple and flexible delegation and display capabilities have been perceived to enhance the manageability of the system quite apart from any security enhancement (if good management and security are indeed separable). Where they have not been offered, management has been reluctant to use the authorization mechanism.

The authorization mechanism, to the extent feasible, should be self-protective. Erroneous, anomalous, or inconsistent authorization data entries should be detected and reported as early as possible—hopefully, at the time of entry and at least at the time they are first used in normal authorization checking. Clearly, the authorization data must be carefully protected against any unauthorized modification.

3.3.4 Authorization Schemes for Confidential Data. Authorization is useful to protect data integrity as well as data confidentiality; however, when the primary concern is for data confidentiality, a relatively simple authorization scheme is possible [DENND 75], [DENND 76].

Access to confidential data can be controlled by tagging sensitive objects with confidentiality labels. These tags or confidentiality labels should be made up so that the same restrictions will be applied to all data with the same labels. A special case of the concept of confidentiality labels is used to protect national security information. The "Confidential," "Secret," and "Top Secret" classification levels as well as other special categories can be thought of as one form of confidentiality labeling. For other types of information, different labels can be used; for example, in a personnel system, the set of labels might include "medical," "psychiatric," "personnel evaluations," "disciplinary actions," and "payroll." Many different labels may be used and several labels may be associated with any one object.

The term "confidentiality label" is used to avoid confusing the general concept with more specific versions of the concept. The general concept is recommended, and it can be tailored to meet a variety of local needs.

Some labels will identify data that is very sensitive, and others will identify data that is less sensitive; for example, top secret data is more sensitive than secret data, and psychiatric data may be considered more sensitive than medical data. On the other hand, the labels need not form a strict hierarchy of sensitivity levels. The authorization scheme need not require a decision about whether medical data is more or less sensitive than personnel evaluations. The decision about who is allowed access to each type of information can be made without determining its relative sensitivities.

Users are authorized access to information by associating a set of labels with each user—thus establishing access privileges for the user. The simple access control rule is that users can access an object only if their set of labels includes *all* the labels that are associated with the information. In some cases it may be desirable to have access privilege labels for terminals and for internal program modules in order to control the information that they can access.

To implement hierarchic relationships between labels, a user who has a certain label should also have any labels that are strictly less sensitive; for example, if a user permitted to access psychiatric information is also permitted to access medical information, then the user's set of labels should include both "psychiatric" and "medical."

The primary advantage of confidentiality labels is that they provide an authorization scheme which is simple enough so that errors in enforcing the scheme are less likely. Delegation is especially simple since it is only necessary to establish confidentiality labels for each user. Furthermore, the same scheme can be applied throughout all associated manual activities. The primary weakness of confidentiality labels is that they are not readily usable to protect data integrity.

The number of distinct confidentiality labels that can be used will determine the flexibility of the authorization system. If there are only a few, centrally administered labels, then administration of the authorization system will be easy, but it will be limited in its flexibility to enforce the principle of least privilege. If there are a large number of labels, and especially if they can be created dynamically, then more flexible control over confidential information is possible.

Confidentiality labels should be carried along with information whenever it is moved from one location to another. (Within a software implementation, confidentiality labels need not be stored with the information as long as they can be determined reliably when needed.) The confidentiality labels should appear on all output, and the authorization scheme should be applied in all the manual information handling activities associated with the application.

3.3.5 Management Considerations. Since damage to the authorization mechanism or data can disable operations, some bypass mechanism or procedure may need to be available so that management has the option to continue operations in an unprotected mode. The designers of authorization mechanisms should not assume that management uniformly believes that system shutdown is preferable to interrupted or degraded protection. Any bypass mechanism is sensitive and dangerous and must be shown to be safe from unauthorized use.

It can be a traumatic experience to transform from a system with little or no authorization control to one that is heavily controlled. In some cases, management simply does not possess the required specific information to create the complete set of authorization data. Thus, a gradual transition is indicated. One approach is to include a capability such that initially, while all checking based on the growing body of authorization data is done normally, authorization "failures" do not cause denial of the requested interaction. Instead, a record of the failure is kept for subsequent analysis. In this way, management can accumulate the appropriate authorization data without disrupting normal operations. As confidence in the authorization data increases over time, the normal authorization failure processing can be used increasingly until the system has been fully converted.

3.4 Journalling

PROBLEMS ADDRESSED:

It is difficult to detect security violations unless there is a record of system events which can be analyzed.

Users are less likely to engage in malicious activities and to make errors in their day-to-day interaction with the system if they know that their activities may be recorded.

Recovery from system failures is aided by journalling appropriate events.

The recording of selected events as they occur within a system, referred to as journalling or logging, provides a basis for identifying and tracing events involved in the processing of data and in the use of computer resources

[BJORL 75], [CARRJ 77], [KRAUL 79], [HASA 77], [MARTJ 73], [HOFFL 77]. Journals of external events in the operational environment can be maintained manually while journals used to record activities internal to the application system must be automated.

One or more automated journals may be employed; the system designers should structure the journals to permit flexibility while minimizing overhead.

Journalling can produce a large volume of data that must be analyzed if it is to be meaningful. The analysis is part of variance detection which can be performed either dynamically as the journal is created or after the fact through a post-processing analysis of the journal. In addition, the journals gather information that can be used to optimize the use of system resources, to reconstruct data bases after system failures, and to audit the system by following the history of a transaction through the system.

3.4.1 Contents of the Journal. From a security standpoint, the ideal journal would include a 100 percent recording of all events relating to data, software, and system resources. From a practical standpoint, such a journal may, in some systems, be out of the question as it could not meet the cost-benefit test[3]. The question of what should be included can only be answered in a general way; the journal must be designed for each individual system and be dependent upon the security objectives for the system and the environment in which the system operates.

In determining what should be journalled, it is important that security requirements be carefully considered and reduced to formal statements. The requirements should be stated in positive terms and include such items as:

- definitions of what kind of data is to be protected and how the system will recognize such data,
- the degree of accuracy that is necessary for various types of data, and
- the definition of who is authorized to access protected data and how the system would recognize an authorized user.

The contents of the journal should be the subject of a comprehensive study that uses the system's mandatory security specifications as a departure point. The results should define:

- the type of information to be recorded,
- the parameters to be applied,
- the analysis to be made,
- the reports to be produced, and
- the retention period of the recorded data.

Selecting the specific data to be captured and journalled is a managerial decision that should be made only after detailed analysis by both users and data processing personnel. Journalling and the associated processing of the journals can impose a performance burden. Consequently, cost must be considered in relationship to the benefits to be gained. The type of information typically included in journals is illustrated by the following:

- nature of the event
  - input and output of data (e.g., opening and closing of file),
  - updating, changing, or modifying data sets of files, and
  - system usage or attempted usage (e.g., logons and logoffs).
- identification of all involved elements
  - people,
  - devices,
  - software, and
  - data (indexes, directories, files, etc.).
- information about the event
  - time and date,
  - success or failure,
  - authorization status of involved elements,
  - transaction numbers,
  - system response,

---

[3] The most common argument against journalling all events is that it is too expensive. In some cases it may be worthwhile to journal all events and retain the journal for only a few days, provided the journal is promptly processed each day after acquisition.

- addresses of items updated,
- contents of items/records before and after creation, update, and deletion,
- programs used,
- results of compatibility and parameter checks, and
- procedural violations.

For automated journals, due consideration must be given to the problems of archiving extensive data for what may be prolonged periods, even years. The ability to easily off-load voluminous journal data, to condense it as much as possible, and to on-load easily the same data for inspection much later in time, perhaps on a different machine complex, is important. These capabilities may also be required for other reasons—especially for internal and external auditing activities.

3.4.2 The Journal as Legal Evidence. The journal can be a valuable indicator for alerting management when an employee is involved in fraud, embezzlement or other misuse of the system. In addition, it may be necessary to use information contained in the automated journals as evidence of the employee's illegal activities. Under the Federal Rules of Evidence, computer output (e.g., automated journals) is admissible in both civil and criminal proceedings if it is determined to be:

- a regularly kept, timely record
- of regularly conducted business activity
- whose preparation has been deemed "trustworthy" by the court.

The first two conditions are relatively easy to establish for a well-run enterprise; the third may present a problem if it cannot be shown that the records and the recordkeeping process itself are reasonably safe from accidental and unauthorized intentional interference. This showing is not only essential to the court's determination of admissibility, it is also an important defense against attacks upon the credibility of the output.

3.4.3 Management Considerations. In journalling, the level of detail monitored, the amount and types of data saved, the length of time the data must be saved, and the complexity of processing the data not only affect the security effectiveness of this tool, but also influence the cost and hardware requirements of the system. Care must be exercised when implementing this tool to prevent the system from being bogged down with journalling to the exclusion of useful work.

The journalling activity is vulnerable to accidental or deliberate events that can endanger the journalling process or the journal contents. Examples include:

- The journal contents may be partially or totally lost in a system crash caused accidentally (e.g., by a power failure to a volatile storage device) or deliberately (e.g., by someone trying to avoid detection).
- The journalling can be disabled or the journal destroyed by a system penetrator.

It is not always practical or possible to protect completely the journalling activity against such events. While it should be protected to the extent possible, management should be aware that the journal may not be available when it is most needed.

The journal produced by an application system will not log activities not performed by the application system; e.g., a file opened outside of the application system.

3.5 Variance Detection

PROBLEMS ADDRESSED:

Fraud, embezzlement and other misuse of the system and its data by authorized users of the system.

Undesirable modification or disclosure of information that remains hidden from management review.

The objective of variance detection is to allow management to detect and react to departures from established rules and procedures that it has determined may constitute hazards. Variance detection acts as a strong deterrent to authorized users abusing their privileges since they perceive the risk of detection to be unacceptably high [MARTJ 73], [CARRJ 77].

Variance detection is useful whenever it is not practical to prevent the undesirable activity by means of an authorization mechanism. In many cases there are tradeoffs between preventing and detecting an undesirable activity with the choice depending on the potential harm that could occur. In other cases there may be no way to determine in advance whether an action should be prevented.

Some suspicious activity can only be identified when the journal discloses a highly unusual pattern of activity. In such situations, the analysis may involve determining what information an individual has accessed over some period of time and ascertaining what the individual could be doing with that information. The complexity of the analysis may preclude prevention even though the activities can later be identified as suspicious. An example would be an employee who obtains a complete listing of sensitive information by piecing together small portions that have been gathered in repeated requests. Assuming that the employee is not authorized to obtain a complete listing, only an analysis of the journal files is likely to uncover the variance.

The mechanisms required to support variance detection are related to mechanisms needed for several other purposes. Recovery, accounting, load-balancing, tuning, and the identification of recurring user difficulties all require some of the same capabilities. The design may be such that one multipurpose mechanism can accomplish all or most of these ends.

3.5.1 Management Inspection of Event Journals. To facilitate management inspection of manual journals, it is important that they be:

- suitably formatted,
- legible,
- complete,
- timely, and
- correct.

For management inspection of automated journals, a post-processing capability should exist that offers both interactive query and report generation functions. This capability, often referred to as static monitoring, requires analytical and retrieval programs that perform functions such as listing:

- all or selected activity initiated from a given terminal or by a given employee,
- transactions of a predetermined type,
- all events falling outside predetermined parameters,
- attempted accesses to restricted files,
- excess resource usage,
- changes made to access privileges of users,
- program detectable software malfunctions,
- abnormal terminations of job runs, and
- requests for unauthorized use or modification of a program, file, or sensitive system data.

Post-processing analysis can perform complex correlations using the journalled information and the logs of system activity produced by the operating system. Patterns often can be determined that direct attention to a particular terminals, user, time of day, etc., and thus assist in ferreting out an error-prone or malicious user.

Reports and other output must be as concise as possible, and specifically pinpoint any unusual actions that occur. Long reports containing large amounts of data may actually decrease detection of security violations as they are tedious and difficult to use. While concise information is desirable, details should be available through use of the journals and retrieval programs.

Management must be able to specify what reports are to be generated periodically. In addition, special report generation functions or an interactive query capability to extract information from the journals are useful. These retrieval programs should support complex Boolean and arithmetic operations upon data elements, numeric values, and statistical data derived from the journals. If management does not have this processing flexibility and power, the likelihood is that the journal will not be inspected regularly; users will realize this; and the deterrent value will be lost.

3.5.2 External Variance Detection Methods. In addition to inspection of the journals, several other approaches can be used to detect variances that occur in the manual procedures and practices associated with the application system:

- observations (visual, electronic, photographic),
- employee interviews and questionnaires,
- undercover operations,
- surprise visits, and
- scrutiny of employee reports for anomalies.

3.5.3 Response to Variances. There must be clearly defined responsibilities to investigate security variances once they have been detected. Since responsibility for security may be divided among many organizations, there is danger that no one will be accountable for security variances when the source of a problem has not been clearly identified. In order to avoid involving senior management unnecessarily, the responsibility for investigating such security variances should be assigned to appropriate organizations. It is reasonable to have the responsibility for investigating security variances depend on the general type of the failure. For example, if a data base becomes garbled with errors, preliminary responsibility may be assigned to the user organization that generated the data unless it is shown that the data integrity was destroyed by errors in programs, operations, data entry, or other functions that are not the responsibility of the user organization. Data confidentiality variances may be assigned to the person who has responsibility for the system of records under the Privacy Act, and ADP availability failures to the ADP operations organization. Responsibility for identifying holes in the overall assignment of responsibilities will naturally fall to the organizations that assume the responsibility for investigating variances.

While direct responsibility for each aspect of security should be assigned to only one person or organization, others should have a clearly defined responsibility to report security variances that they can detect. For example, while source data accuracy is not the primary responsibility of data base administrators, they should be required to report the existence of inaccurate or incomplete records. Similarly, while computer operators may not be responsible for preventing unauthorized access to the computer room, they should be required to report suspicious activity.

3.5.4 Dynamic Monitoring. Some variances can and should be detected in real time so the response can be immediate. Dynamic monitoring is usually integrated with the journalling activity so that suspicious activity is reported immediately.

A fair amount of logic may have to be built into the application system software if suspicious events are to be distinguished from normal processing activities. In general, certain events can be designated as "trigger-events" which cause the monitoring routines to be invoked. The monitoring routines can then examine other factors to determine if the event should be reported or responded to. Flexibility should be provided to permit monitoring of selected events such as all transactions meeting certain criteria, activity initiated from any given terminal, and all or selected transactions initiated by any given user. In addition, monitoring routines can be activated periodically to try to identify unusual values or patterns of activity.

It should be easy for management to establish and change the "trigger-events." To reduce the overhead that is caused by excessive monitoring, only a limited number of trigger events may be used at any one time. The deterrence effect is still substantial as long as management varies the set of trigger events in some random way.

A number of optional reactions to suspected variances can be used; for example:

- real-time alerts to management by the application system software such as a warning bell and a message to a designated console,
- suspension or termination of an offending process with a message to the user that will not arouse concern while the variance is investigated,
- automatic invocation of special monitoring such as complete journalling of all activities and interactive traffic of the offending process, and
- real-time display at a designated console of the full interactive traffic of an offending terminal.

There is some danger that the monitoring software could be used to violate security, and it must be protected from all unauthorized use. Controls must be established to ensure that the monitoring activity is conducted only from secure terminals under the direct supervision of the organizational elements responsible for the monitoring.

3.5.5 Management Considerations. Variance detection is not foolproof, and management should not rely on it as the only safeguard. Nevertheless, variance detection can be very useful to encourage a general awareness of security and to discourage dishonest behavior.

Monitoring of system use can also be employed to identify changes needed to improve efficiency of the work flow in and around the system. This utilization can result indirectly in improved security of operations. If a system user is advised that management has detected a pattern of frequent errors in the user's conduct of certain activities and is offering help, there will be increased awareness on the part of the user that such activities are being reviewed. In this way, monitoring can be productively employed without the necessity of justifying it on the basis of detecting or inhibiting dishonesty on the part of the users.

Clearly, management must be very careful about the way monitoring activities are perceived by the employees. It must be perceived as necessary and even as helpful. If an event is investigated as suspicious and no evidence of wrongdoing is discovered, then the employees must be convinced that there will be absolutely no lasting effects. No-fault variances caused by employees should not be counted nor kept in records associated with them. In most cases, the employees should not even feel a need to avoid repeating the acts that were identified as a possible variance.

If there is even one instance where monitoring activities are perceived by the employees as being used unjustly to harm an employee, the effect on employee morale can be devastating. Unfair usage of monitoring information can even lead to vindictive acts which destroy everything that was being protected by the entire security program.

3.6 Encryption

PROBLEMS ADDRESSED:

> Disclosure and undetected modification of information during its transmission through a potentially hostile environment, including intermediate nodes in a network.
>
> Disclosure of information when tapes or other storage media are lost or stolen.

Encryption [FEISH 73], [KAHND 67], [POPEG 79A], [SHANC 49], [CARRJ 77], [HOFFL 77] transforms data into an unintelligible form, typically by the use of an algorithm and a key. Encrypted data can be decrypted (return to its original form) only by using a companion algorithm and either the same key in the case of a "secret" key system or a companion key in the case of a "public" key system [RIVER 78]. A standard "secret" key algorithm for encrypting and decrypting computer data is specified in [FIPSP 77A]. The key is a very large "secret" number that effectively prevents decryption of the data by anyone who does not know the specific key.

Encryption is useful to protect computer data while it is being transferred between people or devices authorized to have the data and while it is inactive in storage. Encryption protects data from unauthorized disclosure because anyone who does not know the correct key is unable to derive intelligible meaning from the encrypted data. Encryption allows even slight modification of encrypted data to be easily detected because any modification of the encrypted data will result in gross and unpredictable distortions of the data in its decrypted form. Encryption does not protect the data from destruction—in fact, it increases the chances that the data may be destroyed or unavailable because if the key is lost, the data encrypted with that key is lost forever!

The applications that are most likely to need encryption are those that transmit highly confidential data across communication lines. Applications that transmit financial transactions or other critical data may also need encryption if someone is likely to derive enough benefit from modifying the data during transmission to compensate for the risk and the cost of the effort. Encryption of data in storage is an alternative that may be more cost effective than other storage security controls—especially when appropriate support for encryption is readily available.

3.6.1 Encrypting Computer Communications. Encryption is the primary control to protect data from the hazards of communication systems. It can be used to forestall the consequences of electronic eavesdropping, intentional modification, misrouting, introduction of spurious messages, spoofing (falsifying the identity of the transmitter), playback of old messages, and for the authentication of senders and receivers and the protection of acknowledgements that the message was received or not received correctly. The primary problems when using encryption for this purpose are to ensure that the encryption does not interfere with communication protocols and to guarantee that the correct key is available to all authorized recipients of a message—and to no one else. For details, see [COLEG 78], [BRAND 78].

3.6.2 Encrypting Off-Line Storage. Encryption of data on tapes and other off-line storage media is an effective way to protect the confidentiality of the data without having to maintain rigorous physical control over the data storage media. Again, the primary problem is to ensure that the correct key will be available to decrypt the data when it is needed—and that it will not be known to anyone else. In general, the keys should not be stored on the computer because then they would be subjected to all the hazards applicable to computer data.

Encryption protects the data if the physical storage media are lost or stolen; however, it may not protect the data from unauthorized use through the computer system. Since the data must be decrypted whenever it is to be used, the encryption will not protect the data from anyone who can masquerade as a legitimate user of the data.

3.6.3 Encrypting On-Line Files. Since encrypted data cannot be processed until it has been decrypted, encryption of frequently used data will not be very efficient unless the encryption is supported by special features in hardware and in the operating system.

For encryption of on-line data to be useful, the keys may have to be stored on-line. Even if the keys themselves are always encrypted except when they are within the encryption device, it is still necessary to verify the identity of anyone requesting a key, to determine whether the key has been authorized, to journal the usage of a key, and to detect any variance in the usage of a key. In short, the primary controls discussed previously will still be necessary to protect the keys; however, encryption can reduce the scope of the protection problem because it is no longer necessary to ensure rigorous control over those areas of computer storage that contain only encrypted data.

3.6.4 Management Considerations. The cost of hardware that implements the encryption algorithm can be expected to decrease rapidly throughout the early 1980's. As more experience is gained in the use of encryption, the costs of planning for its use and of managing the distribution of keys will also decrease.

Before hardware and software implementations of the standard data encryption algorithm can be procured for Federal use, manufacturers must obtain a certificate of validation for their product from NBS.

If encryption is to be used at all, it should be used properly. A poorly planned use of encryption, or a poor system for managing keys will give a false sense of security.

[illegible] SELECTION OF CONTROLS THROUGHOUT THE SYSTEM LIFE CYCLE

The selection of appropriate security controls depends not only on the sensitivity of the data and the degree of harm that could result from improper actions, but also on the vulnerabilities that arise in the environment where the application operates. Thus far, no standard methodology for selecting controls has been proposed for implementing applications in an ADP environment. However, a good discussion of the required considerations for selecting appropriate security measures is found in [ORCEM 78].

While the selection of controls is still an art based on experience and good judgment, it is possible to make this selection if it is not overly constrained by poor planning for security. When compared with the level of security that is often found in computer applications, dramatic improvement can be achieved as long as security controls are selected by experienced personnel who:

1. are familiar with the vulnerabilities of the entire application system—including both automated and manual procedures,
2. can take a system-wide view of the application and have the freedom to locate controls at the most appropriate points, and
3. understand common application system control objectives and techniques [HASA 77], [MAIRW 76], [BIGGC 80], [AACO 78], [AICPA 77], [EDPAF 77], [SGCCA 75B].

These guidelines will be of direct help in achieving conditions (2) and (3). Each agency has to rely on its own personnel to fulfill the first condition and to select the appropriate controls. The remainder of this document describes how condition (2) can be achieved as long as security is considered at each step of the application system's life cycle. While it is common practice for system developers to think of system functionality first and to delay security concerns until later, many opportunities to include effective controls are lost if security is not considered early.

Basic guidance to fulfill the third condition is contained in section 3. Additional guidance about enforcing security controls during development and operations is contained in sections 6 and 7.

[illegible] Application System Life Cycle

The life cycle of a computer application consists of three identifiable phases—initiation, development, and operation [FIPSP 79A], [FIPSP 76A], [FIFED 77]. After some period of operation, the system will have to undergo an expansion or revision, and the life cycle is then repeated. It is desirable for the life cycle of the application to be independent of the life cycles of the computer hardware, of the operating system, of the computer facility, and of associated applications.

INITIATION PHASE. The initiation phase establishes the objectives and general requirements of the computer application. System planners consider alternative approaches for a target system. Based upon feasibility studies and cost-benefit analyses of the potential solutions, a decision to proceed with the development of a specific system is reached. If security is not taken into account during this initial planning, feasibility studies and cost-benefit analyses may be seriously distorted.

DEVELOPMENT PHASE. The development phase consists of four stages—definition, design, programming, and testing. While these are logically independent stages, in practice they may overlap substantially. Most decisions about which security controls to include must be made in an orderly way during these stages. In addition, there are software development controls that must be applied during these stages to ensure that the desired level of security is achieved.

[redacted] The operational phase begins once the system has been accepted by its intended users, and they become dependent on it to fulfill their organization's mission and responsibilities. Operational security controls must now be enforced during day-to-day operations.

4.2 Improving Security for Existing Applications

With some minor modifications, the life cycle approach to security applies to computer applications that already exist. When the system is already in its operational phase, the following three steps are recommended:

RISK ANALYSIS. It is best to begin with a thorough understanding of the risks and exposures in the present system. This should be accomplished by a risk analysis following the methodology of [FIPSP 79B]. The risk analysis will indicate how urgent and how extensive the revision must be.

IMPROVE OPERATIONAL CONTROLS. The risk analysis will usually uncover some vulnerabilities that can be corrected by improving the operational controls described in section 7. The improved operational controls will be necessary even after the system is revised to improve security, and they will usually reduce the risk significantly while more thorough improvements are planned and implemented.

REVISE THE APPLICATION SYSTEM FOLLOWING THE LIFE CYCLE APPROACH. Revisions of the computer application should pass through the phases of initiation, development, and operation of the revised system. The guidelines in each of the following sections should be used for the revision—in an abbreviated and less formal way when the revision is minor.

When possible, it is best to integrate security improvements with revisions that may be needed for other purposes to minimize making repeated small modifications which are likely to make a computer application more complex and less reliable. This applies both to the software and to manual procedures. Even though it may be desirable to implement the revision as a set of small evolutionary changes, the changes should be planned in a coherent way to prevent the degeneration of the system.

[redacted]

The need for effective internal auditing [USGAO 72], [RCBCG 79], [USGAO 74] in Federal agencies has been recognized by the Congress in a number of laws, particularly the Budget and Accounting Procedures Act of 1950 and the Inspector General Act of 1978. An internal audit is an independent appraisal activity within an organization for the review of operations as a service to management. It is a managerial control which functions by measuring and evaluating the effectiveness of other controls. The responsibility of the internal auditor includes verification and evaluation of controls, standards, and data processing results [redacted]

1. the accuracy and reliability of the data maintained on or generated by an ADP system;
2. the appropriate protection of the organization's information assets (including hardware, software, and data) from all significant anticipated threats or hazards; and,
[redacted]

With the growing dependence of organizations on technically complex ADP systems, it has become increasingly clear that in order to perform an internal audit and, in particular, a computer security audit, the auditor has had to modify, extend, and supplement traditional audit approaches to manual systems and to broaden the scope of the audit to include review during all life cycle phases of computer applications. As evidence of this, GAO has recently published three supplemental standards for auditing computer-based systems [USGAO 79] which require that:

> "1. The auditor shall actively participate in reviewing the design and development of new data processing systems or applications, and significant modifications thereto, as a normal part of the audit function.
>
> 2. The auditor shall review general controls in data processing systems to determine that (A) controls have been designed according to management direction and legal requirements, and (B) such controls are operating effectively to provide reliability of, and security over, the data being processed.
>
> 3. The auditor shall review application controls of installed data processing applications to assess their reliability in processing data in a timely, accurate, and complete manner."

These standards reflect the current belief of the audit community [RUTHZ 77], [RUTHZ 78], [HASA 77], [RCBCG 79] that adequate and effective security of computer applications can be achieved only by audit review and evaluation throughout the development and implementation of an application. This recommended practice ensures that:

- adequate controls are built-in to the application;
- the capability to adequately track events during systems operation is provided to facilitate future audit reviews; and
- accepted accounting principles are complied with for financial applications.

[redacted]

Decisions made during the initiation phase [FIPSP 79A], [BIGGC 80] will determine:

- The role of the ADP system and the degree to which the organization will rely on it to fulfill its mission.
- The possibilities for effective manual review of results from the ADP system.
- The nature and sensitivity of the data to be stored in automated form.

Security concerns must not be neglected as these decisions are being made.

In general, the relationship between the computer application and associated manual activities should be planned so that errors in one activity can be detected by the other. The computer application should be able to detect erroneous or fraudulent manual activity; and, when possible, manual activity should be planned so that critical output from the automated system is reviewed before final action is approved.

The initiation phase for revising an existing application is different from the initiation phase for a new application. When controls must be added to an existing system, cost constraints are different and the planners will find that they have fewer viable options open to them. Some controls may be reasonably simple additions to the existing system. Other controls will require major changes in the software design or in the external environment of the application system; and implementation of these controls will be costly unless a major revision of the system is needed in any case.

Before the planners decide to limit the scope of a revision, they must be sure that the easy controls are the controls that are needed. Money spent on simple enhancements might be wasted if the most serious vulnerabilities cannot be corrected without a more thorough overhaul of the system. The security feasibility questions of section 5.1 should all be asked as part of the risk analysis for an existing application. Some difficult decisions will have to be faced if the answer to any of these questions is not a clear "yes." Negative answers to these questions usually mean that a major revision of the computer application should be planned; however, in some cases it may be better to plan a sequence of revisions that will culminate in a system with a coherent and effective set of controls.

For new systems, there is a serious tendency to neglect security and integrity concerns until the nature of the system is clearly defined. Unfortunately, this can lead to systems whose basic characteristics are inconsistent with appropriate security objectives. The following two sections will help to identify security problems that are inherent in a proposed system. These sections should be used as part of the feasibility studies and the cost-benefit analyses, respectively.

[redacted]

PROBLEMS ADDRESSED:

Unrealistic plans for a computer application may result in security problems for which there is no cost-effective solution.

System designers and implementers will not perform miracles—it cannot be taken for granted that they will be able to build inadequate security controls. The five questions below cover the major concerns about security feasibility. [redacted]

[redacted]

Source data will not be perfect, and additional inaccuracies may be introduced before the data comes under the control of the ADP system. Even if the overall error rate is very low, serious harm can result from a few errors if they have been deliberately introduced or if they involve critical data.

If the proposed system will use the data in the same way as an existing (manual or automated) system, then it should be relatively easy to evaluate the potential harm from inaccurate or incomplete data. If the proposed system uses new data sources or uses present data for new purposes, then a more serious study is needed.

In an automated system, the lack of human review of the data leads to some increased dangers; however, the ADP system can also contribute to improved data quality to the extent that automated data validation can be used to detect and reject inaccurate and incomplete data. The effectiveness of automated data validation techniques in identifying harmful inaccuracies is very application dependent and must be evaluated for each application system.

5.1.2 User Identity Verification. Can users of the system be adequately identified and authenticated so they can be held accountable for their actions [FIPSP 77B]?

If former systems did not require users to identify themselves, then a major change in organizational procedures may be necessary. Furthermore, can outsiders and other unauthorized individuals be prevented from using the system? Rigorous control over who uses the system is especially important if the system is to allow access from dial-in telephone lines.

5.1.3 Restricted Interfaces. Are the user interfaces to the system sufficiently restricted so that adequate security is feasible [HOFFL 77], [CARRJ 77]?

If there will be a large number of authorized users, then the proposed system should restrict most users to limited forms of access to the system. Data accuracy will be very difficult to maintain if many different people can update a master data file. Even query rights to a data base should be examined for consistency with privacy and security requirements before they are planned for widespread use—especially by persons from outside the immediate organization served by the application. Since current operating systems do not prevent a maliciously written program from bypassing all available security controls, high-risk applications should be planned so that a very limited number of people will be allowed to submit programs to the computer facility that is running the sensitive application.

5.1.4 Separation of Duties. Do the boundaries between ADP and related manual activities provide maximum separation of duties and independent review [IBMCO 76]?

Errors are more likely to be detected and frauds deterred when two or more independent people are involved in all important actions. Security is easier if the department supported by the ADP system maintains controls over ADP and related manual activities. In most cases, data collection, initiation of transactions, authorization of expenditures, and physical control of assets can be maintained by the user department[4].

5.1.5 Facility Security. Is the proposed processing facility adequately secure [FIPSP 74], [RCBCG 76C]?

Selection of the computer facility need not be done before the design stage; however, if there are any assumptions or constraints on the possible facilities, they should be reviewed for consistency with security. Are physical security measures, operational controls, personnel and administrative procedures, emergency plans, and backup and recovery procedures at the proposed facility adequate for this application? Are the availability requirements of this application consistent with the performance of the proposed facility? For high-risk applications, is it acceptable to run the application concurrently with the other applications at the facility?

5.2 Initial Assessment of Risks

PROBLEMS ADDRESSED:

If security risks are not considered, cost-benefit analyses may favor a system plan with unnecessary exposures to traumatic failures.

A risk analysis [FIPSP 79B], [LOSOT 79], [WONGK 77] determines an annual loss expectancy which should be taken into account in the cost-benefit analysis. Furthermore, a risk analysis is a useful first step toward estimating the maximum tolerable cost of appropriate controls.

---

[4] The user department will need some redundant manual records if its controls are to be effective. These redundant records can also be used for backup and for auditing, however, paper copies should be kept only if it will be practical to use them when needed or when such retention is required by law or regulation.

When planning revisions for an existing application, a risk analysis for the proposed revision can be obtained by adjusting the risk analysis for the current system. The remainder of this section applies primarily to new applications or to applications undergoing major revisions.

A thorough risk analysis, including safeguard selection, should be performed at the beginning of the design phase to assure that appropriate cost-effective security controls are integral to the system's design. The initial assessment of risk should only estimate the damage that could result from major failures that might occur in spite of controls. With appropriate controls, such failures should be rare; however, some possibility of failure will remain. When possible, plans for the computer application should be revised to minimize the potential impact of such failures.

A risk analysis involves an estimate of the impact of a security failure and an estimate of its frequency of occurrence. The product of these two factors gives an annual loss expectancy. This annual loss expectancy from major failures, derived from an initial risk analysis, should be included when considering the cost-benefit tradeoffs between alternative systems.

5.2.1 Impact of Major Failures. For the purposes of this initial assessment, the impact of a security failure can be described in any convenient terms—dollar value of loss, extent of inconvenience or hardship to individuals, lives lost, or degree of disruption to the national economy or national security. As long as major impacts are not being overlooked or minimized, it is possible for potential impacts to be identified quickly.

The impact of at least the following failures should be assessed for each major body of information that is to be processed by the proposed system.

- *Inaccurate Data.* Data (or programs) become corrupted with errors, but the system continues to function while producing erroneous outputs. Estimate the potential impact of the erroneous actions that might result assuming only that the output of the ADP system is not so obviously out of line with reality that the errors would be noticed. Consider both the impact of a few very serious errors and the cumulative effect of many small errors.

- *Falsified Data.* An individual manages to falsify information in order to gain some advantage. The falsifications are limited only by the fact that they are subtle enough so they are not detected manually. Estimate the total impact that could occur over an extended period of time. Example: In a funds disbursing system, the potential impact might be limited to some percentage of the funds being handled. This limit assumes that losses over that percentage would be noticed based on budget projections and that such losses would eventually be corrected.

- *Disclosed Data.* Sensitive data in the system becomes available publicly or to certain individuals. The unauthorized disclosure of the data is not necessarily discovered.

- *Lost Data.* Data (or programs) used by the application system are destroyed or corrupted. Backup versions are nonexistent or not usable, and the data must be reconstructed manually. Estimate the impact of losing the data. If manual reconstruction is obviously not feasible and if backup in depth is anticipated, then estimate the impact of the inaccuracies and the temporary unavailability that would result while recovering from an old backup copy on the assumption that all recent backup copies had been destroyed.

- *Unavailable Data or Services.* The computer hardware or related equipment in the computer facility is disabled and the system is not available until it can be brought up on another facility.

5.2.2 Frequency of Major Failures. An estimate of the impact of a major security failure is not meaningful without some estimate of how frequently it might occur. Unfortunately, during the initial planning for a system, it is not possible to estimate the frequency of a major failure by evaluating the effectiveness of the controls because those controls have not yet been designed. Nevertheless, it is possible to make a very rough estimate of the likelihood of a major failure. If the proposed system is generally comparable to other computer applications, then a major security failure of the sort described above can be estimated to occur about once in a hundred years. This simple estimate is based on the following line of reasoning:

> Available security controls (if they are managed properly) can prevent *major* security failures from recurring as frequently as once every 10 years. On the other hand, any ADP system has several vulnerabilities against which there is little defense; most systems can be manipulated by any one of several individuals who are in a position of trust—programmers, those responsible for security, the computer operators, and others. There are enough instances of major security failures in computer applications so that an expected frequency of once in a thousand years is very optimistic. The estimate of once in a hundred years is only intended to be accurate to within an order of magnitude.

This estimate for the frequency of a major security failure can be used to give some perspective on the estimates about the impact of a failure and to compare the risks of an automated system with those of a manual one. Since the estimates of frequency of occurrence used during the initiation phase will be the same for all automated systems, only the estimates of the impact of a major failure will be useful to differentiate between alternative plans for automated systems.

6. BUILDING IN SECURITY DURING THE DEVELOPMENT PHASE

[redacted] PSP 76A]. This section describes the security concerns that must be dealt with at each stage.

In addition to the specific security concerns described in this section, the security and integrity of an application system are strongly affected by the overall quality of the development effort [FIPSP 76A], [FIFED 77], [BIGGC 80], [MCCAJ 77A], [MCCAJ 77B], [MCCAJ 77C], [BOEHB 73B], [MILLH 77]. Effective management of the development activities is one way to improve security while at the same time reducing costs. If the software development effort is poorly organized and depends mostly on debugging and testing for its quality control, then software controls are not likely to be effective, and development costs are likely to be excessive. The following guidance specific to security should be used in conjunction with the more general guidance on managing software development in [FIFED 77].

6.1 Definition of Security Requirements

PROBLEMS ADDRESSED:

Many security problems can be traced to a poorly thought out security plan or to an inadequate definition of what the software was supposed to do.

The decisions about what security provisions are needed and to what extent they are to be enforced by software must be made and documented before the software development can proceed. The documentation of these security requirements constitutes the security specification called for in OMB Circular A-71, Transmittal Memorandum number 1, paragraph 4.c(1). This security specification may be incorporated into the functional requirements document and the data requirements document [FIPSP 76A], or it may be an independent document.

Programmers should not be expected to identify the security concerns that arise from the user's application. Security requirements should be defined by personnel from the organizations that are to be served by the application system as well as by computer professionals who have experience developing similar systems. Internal auditors, who are responsible for evaluating the adequacy of controls, should be called upon to review the security requirements. The proper completion of security requirements definition is more important and more difficult than is usually anticipated.

The term "requirements" can be used at many different levels. There are very general requirements for any government system to protect the rights of individual citizens and to manage government resources prudently. More detailed requirements come from laws, policy directives, and standards. The requirements defined at the beginning of the development phase are even more detailed and include everything that the users and other responsible parties want to require from the software system.

These requirements should be expressed in a way that allows the software designers to choose the best way of implementing them. Selection of software modules, algorithms, and data structures should be left to computer professionals who are competent to make those decisions; however, the requirements should define all details about the intended application which are not in the primary domain of knowledge of computer professionals. For example, in a financial accounting system the cross-checking of records that is part of standard accounting practice should be a requirement since it is more in the domain of knowledge of accountants than of computer professionals. Similarly, journalling requirements should specify the type of information discussed in section 3.4.1 rather than the algorithms and data structures for implementing the journals.

Some security controls can be enforced either by software or by physical and administrative procedures. For example, data integrity can often be checked either by human review or by automated bounds and consistency checks. When controls can be implemented in software, there are two reasons why this is preferable:

- Once controls have been automated, their continuing cost is usually lower than if they are enforced manually.
- Automated controls will be more consistently applied.

Unfortunately, the advantages of software controls can be lost if they are not implemented carefully. Since software does not adapt to unforeseen circumstances, there is considerable danger if software controls are incomplete or if they can be bypassed with unacceptable probability. Furthermore, if they are added after much of the application software is already designed or operational, the difficulty of modifying software can make their implementation quite costly—and can also lead to compromises that affect their effectiveness and reliability.

The following steps can be used to define security requirements and arrive at a detailed security specification:

6.1.1 Application System Interfaces. Identify each job function which is related to the application system. Consider each job function which relates to the application in a different way as defining a different interface to the system. Define the nature of the interaction between each job function and the system. Also identify and define any interfaces to other automated systems. Include all job functions (or other automated systems) that are to be supported by this system—even if the people performing those jobs only receive reports from it. Also include all job functions (or other automated systems) that supply information to the application system or that support its operation. Be sure to include critical job functions such as:

- Source data collection,
- Input preparation,
- Data entry,
- Output dissemination,
- Data base administration,
- System security planning and control,
- Internal audit,
- Application program maintenance,
- Archival or backup data storage,
- Computer operations, and
- System programming.

6.1.2 Responsibilities Associated with Each Interface. For each application system interface, define the responsibilities of the individuals who interact with the application system through that interface. Identify the constraints on the use of each interface that must be enforced if security is to be preserved[5]. Consider the likelihood of errors occurring in the use of the interface and identify their consequences. Consider the consequences of deliberate misuse of the interface. Without appropriate controls, deliberate misuse can even cause the execution of functions that are entirely unrelated to the intended use of that interface. Identify the management and administrative controls that will be available to ensure that the interface is used properly. Are the requirements on those using the interface and on their managers realistic? See [FDICO 77] for a general but very extensive example.

6.1.3 Separation of Duties. Examine the interfaces to ensure that security exposures will be minimized even if an interface is misused. Ideally, any action that could result in serious harm should be checked or approved from an independent interface. For example, no one individual should be able to trigger a disbursement of funds without some effective form of independent checking or approval. (See sec. 7.2.2.)

Well-planned separation of duties can greatly reduce the life-cycle costs for the background investigations that are required by OMB Circular A-71, TM1, paragraph 4.b. When there is effective independent checking on the actions of an individual, the level of personnel screening can be greatly reduced. Furthermore, independent checking may be more effective than background investigations to deter fraud.

6.1.4 Sensitive Objects and Operations. Identify the data objects to be processed—include input data, stored data, and output data[6]. Determine the sensitivity and asset value of the data objects. Identify the operations or functions that users will perform on this data. Define security requirements for each data object (or group of related data objects) with respect to the objectives of data integrity, confidentiality, and availability. Define requirements with respect to each objective to protect against both accidental and deliberate threats. If some operations such as updating a critical value are significantly more sensitive than other operations on the same object, then distinguish the different operations to be performed on each data object.

---

[5] While this paragraph assumes that a person is on the other side of the interface, similar comments apply when the system interacts with other automated systems. It is usually poor security practice to assume that other automated systems are trustworthy. Whenever possible, it is better practice to determine the degree to which they are untrustworthy or to take the necessary action to make them trustworthy.

[6] The data objects considered at this point are objects seen by the end users, not all the files or data sets as seen by a programmer.

6.1.5 Error Tolerance. Determine the application's error tolerance by taking into consideration the expected reliability and validity of the data and the intended objectives of the application. For example, funds disbursement or electronic funds transfer systems may have a low tolerance for data errors since such errors directly translate into dollars. Real time control systems such as air traffic control have virtually no margin for error since human lives may be lost. On the other hand, some management information systems, particularly those used to predict future demands and resource requirements, may not be as susceptible to errors in data since a certain degree of unreliability is expected in the data, and since statistical estimating techniques are often used which account for variations in the quality of data by generating ranges or bounds for expected results. The application's tolerance for errors and the requirements for maintaining error levels to within acceptable tolerances must be defined in the security requirements.

6.1.6 Availability Requirements. If serious harm could result because the system is not available for use, then availability requirements must be defined. Availability requirements which are more rigorous than those typically supported by a well-managed computer installation are likely to be expensive. Depending on user needs, requirements can be stated in terms of a maximum length for a service interruption or in terms of its frequency of occurrence or as a combination of both factors.

6.1.7 Requirements for Basic Controls. For each application system interface and/or each data object, define detailed requirements for each of the basic controls from section 3.

6.1.8 Management Considerations. The requirements should not contain phrases such as "and so forth" or "and the like." The personnel doing the requirements specification are in a better position to fill in the details than the programmers will be.

All statements in the requirements specification should be specific enough so that tests can be designed that will tell whether the requirement is satisfied. For example, the statement "Data entry personnel should not be able to enter fraudulent payment vouchers" may be a very general requirement, but it is not specific enough to be a useful requirement on the software designers. Software requirements would have to specify the characteristics which can be used to identify a payment voucher as potentially fraudulent. The software requirements specification should be detailed enough to become the basis for the tests to be used during system integration testing.

While requirements definition logically precedes all other development activities, in practice there is often a substantial overlap, and many requirements must be modified or clarified during the remainder of the development effort. Management must establish procedures so that the requirements can be modified in an organized and carefully controlled way. The security specification should be kept current throughout the entire life cycle of the application and no changes to the application system should be permitted unless they either do not affect the security specification or have been approved and entered as an official modification to it.

Security specifications should be reviewed by all organizations that will be involved in the use or operation of the application. For any sensitive application, they must be reviewed and approved by the party responsible for security and by the organization's internal auditors.

6.2 Designing for Security

PROBLEMS ADDRESSED:

Inadequate security which is difficult to improve without a major redesign of the system.

Excessive costs for administrative controls that are needed only because the system design exposes critical code and data to unnecessary hazards.

The design stage is the time to make detailed decisions about how the controls of section 3 will be implemented. Nevertheless, designing for security is not simply a matter of appending controls onto a system design. Security concerns must permeate the entire design phase.

It is not necessary for every design decision to take the most secure approach. There are usually a variety of ways to achieve an adequate level of security, and it is only necessary that security be given an appropriate weight in each design tradeoff. Often a few basic design decisions such as those described below can dramatically reduce the vulnerabilities of the application and make security much less of a constraint in later design decisions.

A primary source of security problems is an excessively complex design which cannot be implemented correctly, cannot be maintained, and cannot be audited. While high performance goals may be an excuse for very complex designs, when security is a concern it is especially important to follow the age-old maxim "Keep it simple." It should be noted that performance and simplicity are not mutually exclusive design features.

For any sensitive application, a thorough risk analysis [WONGK 77], [LOSOT 79], including safeguard selection, should be performed at the beginning of the design phase to assure that appropriate cost-effective controls are integral to the system's design. The guidance in [FIPSP 79B] can be tailored for a risk analysis of an application system design.

Designs for each of the basic controls were discussed in section 3. The ideas described here apply to the overall design effort.

6.2.1 Unnecessary Programming. Terminals should be interfaced with the application system so as to minimize the danger that users can get unneeded programming capability. Users who can execute their own programs usually have the potential to bypass any security control. Operating systems attempt to prevent this, but their controls usually can be bypassed or avoided by clever user programming. While most users do not have the knowledge required to break security in this way, it is a poor security defense to rely on the user's lack of ingenuity.

One should not expect an absolute guarantee that programs cannot be entered from a terminal. A trap inserted almost anywhere in the code of the application or of the operating system and triggered by some designated event could cause data entered from the terminal to be executed as a program. Thus, the effectiveness of this control depends on the integrity of the program developers and of others who have had an opportunity to embed traps in the code; nevertheless, an attempt to reduce the number of people who can enter programs is important because it reduces an otherwise almost uncontrollable exposure.

6.2.2 Restricted User Interfaces. User interfaces should be tailored as specifically as possible to fulfill the user's requirements. Unneeded flexibility makes it more difficult to train users and to get them to accept the system. Such flexibility also hurts security. The greatest danger occurs when users are given unnecessary access to a general purpose programming language; however, even for a generalized query and update language, it may be very difficult to analyze all the possible ramifications on security for all the possible user actions. Once the users' actual needs are understood, interfaces should be designed to meet these needs as simply as possible with no unnecessary capabilities that complicate things both for the users and for the security analyst.

6.2.3 Human Engineering. To preserve security and integrity, user interfaces must be designed so they are easy to understand and use. This can forestall many user errors, and it decreases the chances that users will neglect or bypass controls which they view as cumbersome and annoying. This task is difficult and deserves careful thought.

System responses to the user must be clear and presented in terms of the user's view of the information—not in terms of internal details intelligible only to a programmer. Error messages should be tactful and helpful.

The user should not be required to enter excess information or answer questions repeatedly. When possible, appropriate security should be provided by default options. On the other hand, user requests that could have unusually serious effects should not be executed without having the system explain its interpretation of the request and ask the user for confirmation.

There should be a user manual for each user interface. When possible, help in using the system should be available on-line—a user should be able to type some standard request such as "help" and receive a description of the actions that are valid in the user's current context.

Human engineering may add to design costs, but if it is done effectively, it should decrease life-cycle costs.

6.2.4 Shared Computer Facilities. It is easier to protect the code and data of the application system if it does not share computer facilities with other applications. It is especially useful to exclude all program development activities from the machine that runs the application—including the development and maintenance of the application's programs themselves. For applications that run constantly—and especially for very sensitive applications—it is often cost effective to run the application on its own computer(s). In terms of simple hardware performance, small computers are now at least as cost effective as larger machines; however, the costs for system software, telecommunications, peripheral equipment, and facility management must also be considered before making a choice between dedicated small machines and shared large machines.

6.2.5 Isolation of Critical Code. The code and system data that is critical to security should be well identified so it can be more easily audited and protected. When possible, security controls should be isolated in modules that have few interactions with the rest of the application software. This makes it easier to audit these modules and protect them from unauthorized modifications during operations. Sometimes automated controls can be used to protect these modules:

- Checksums on the object code can be used to try to detect unauthorized changes.
- Hardware protection states or protection domains can protect code and data critical to security [LINDT 76C].
- If the security-critical code always resides in a fixed area of memory, then read-only memory can be used.
- Recent experimental languages, called type safe languages, may soon be available. Compilers for these languages can protect modules and their data from unexpected interactions with other modules [LINDT 76A], [WULFW 76], [LISKB 75], [LAMPB 77B], [AMBLA 76], [AMBLA 77].

Identification and partial isolation of critical code and data are reasonably easy; however, rigorous isolation is more difficult than it sounds. Without very careful planning, all system code and data will end up being relevant to security because errors or deliberate traps elsewhere can still cause security failures. This other software includes:

- The operating system and other software that supports any of the security functions.
- All parts of the application system needed to guarantee that the security controls are invoked at the appropriate times.
- Compilers and other software used to develop and maintain any security-relevant software.

When the code relevant to security is rigorously isolated from the bulk of the software, it is called a security kernel. Security kernels that protect data from unauthorized disclosure are feasible, but they require specially designed operating systems [WULFW 74B], [BRINP 70], [ARMY 78], [BERST 79], [DODKS 78], [FEIER 79], [LINDT 76C], [MCCAE 79], [POPEG 78A], [POPEG 79B], [SCHIW 75], [WOODJ 77].

6.2.6 Backup and Recovery. With appropriate contingency planning (see sec. 7.6), the services of a computer application can usually be restored within a few hours after a failure. If availability requirements are more rigorous than that, then automatic backup and recovery mechanisms may need to be included in the application software. While such mechanisms are beyond the scope of this guideline, several of the references and additional readings listed at the end of this document contain discussions of this area.

6.2.7 Use of Available Controls. The operating system and the facility management may already provide a variety of controls such as:

- User identity verification.
- Authorization for access to system files.
- Journalling of operating system activities.
- Backup and recovery procedures.

While the application system usually needs to supplement these controls, available controls should be used to the fullest extent possible. In too many cases, controls are needlessly reimplemented because controls that are already available are not understood or not utilized.

Since the security controls of an operating system are not absolutely reliable, the application system should use some data integrity checks to try to identify whether critical data has been altered; however, in general, there is no reason to believe that controls implemented in the application system will be any more reliable than those already provided by an operating system.

6.2.8 [redacted] The design review [FAGAM 76A], [FAGAM 76B] is the last reasonable opportunity to identify weak points in the security plan. Omissions or inadequacies in the security design which are not identified at this point will require costly software modifications.

The entire security plan for the application should be reviewed by a group of experts who were not part of the design effort. It is especially necessary to review all decisions which cannot be easily altered during the expected life cycle of the application.

6.3 Programming Practices for Security

PROBLEMS ADDRESSED:

Programming errors that affect security, especially flaws in the implementation of security controls.

Fraudulent additions or changes to the application code by programmers.

Programming practices [WIRTN 73], [DAHLO 72], [WEING 71], [VAND 74], [KERNB 74], [YOURE 75] that are based on the best software engineering methods are essential to security since programming errors are a common cause of serious losses. Furthermore, programming errors or deliberate traps inserted into the code can often be exploited to carry out a fraud or embezzlement. Since program errors and traps are difficult to detect, the most effective approach is to prevent their occurrence.

During the programming stage, the following practices are recommended to enhance the security of the application system.

6.3.1 Peer Review. After a programmer has completed a section of the code, review by one or more peers is requested. Typically, the reviewers will verify not only that the code does not contain any security errors but also that it:

- satisfies all design specifications,
- is efficient, and
- is easily maintainable.

Peer review is most effective when at least one other programmer is expected to review the code to the point of understanding it completely and being made equally responsible with the original programmer for any errors that remain. This is the most effective technique for preventing a programmer from making fraudulent modifications to the code. Peer review is most practical when programs are coded using programming methodologies such as those described in [FIFED 77].

6.3.2 Program Library. The program library catalogs and controls access to all versions of program modules as they are being developed. These control functions can be carried out by either manual or automated means.

The program library can provide the following types of security controls during the programming stage:

- permit only authorized persons access to program modules,
- record all accesses (especially modifications) to program modules,
- associate control data, such as record and byte counts, with program modules to facilitate detection of changes, and
- enable comparison of current versions of modules with previous versions to identify code that was changed.

More rigorous controls are needed as the program modules near completion, especially once review and testing has begun.

6.3.3 Documentation of Security-Related Code. Program documentation is needed during any software development; it is especially necessary for security if software security controls are to be reviewed for effectiveness and if they are to remain effective after software maintenance activities. Security-related modules or sections of code must be clearly identified and completely documented. By security-related code is meant:

- code that implements security controls;
- code that performs critical processing (e.g., check disbursement, real-time control); and
- code that has access to critical or sensitive data during its execution.

6.3.4 Programmer Association with Operational System. When possible, programmers should not be in a position to receive benefits from the system when it becomes operational. The temptation for programmers to make unauthorized modifications or insertions to the code will be greater if it will be easy for them to capitalize on these insertions at a later time. Programmers excluded from access to the operational system can still conspire with someone else to receive a benefit, but the conspiracy increases their risk of being detected.

6.3.5 Redundant Computation. Critical computations can be checked by redundant processing to verify the correctness of the result. The effectiveness of such checks depends, to a large extent, upon the programmers' experience and expertise and on the programming practices being used. Examples of local redundancy checks include:

- recalculation of a critical result by an alternate method;
- checking a calculated result for reasonableness and consistency with other data items; and
- examining extra attributes in a retrieved data item to assure that the data item found was the one that was searched for.

6.3.6 Program Development Tools. The choice of the programming language and of other programming tools can enhance security by increasing the reliability and correctness of the final products. Proper selection and utilization of such tools will help prevent programming errors from entering the source code. Some specific program development tools include:

- *High Level Programming Languages.* Programming languages are especially useful if they support structured control flow, extensive data definitions facilities, strong type checking, restricted scopes for program variables, and well-defined module calling conventions. Compilers for such languages can do extensive checking to identify program errors [WULFW 76], [LISKB 75], [LAMPB 77B], [AMBLA 76], [AMBLA 77].
- *Preprocessors.* Many of the advantages of high-level programming language can be accomplished through a preprocessor. A preprocessor accepts programs written in an improved dialect of a language and translates them into the language. Preprocessors can be used to:
    - eliminate some of the more restrictive conditions in an existing language (e.g., allow structured flow in FORTRAN programs), and
    - provide automated quality control by checking that program modules meet the coding standards of this project.
- *Other Tools.* Program development tools such as those that reformat source code, produce cross-reference listings, and aid debugging are useful to help programmers manage the complexities that they must deal with.

6.4 Test and Evaluation of Security Software

PROBLEMS ADDRESSED:

Errors and omissions that affect security, especially flaws in the implementation of security controls.

Test and evaluation [LARSR 75] attempts to demonstrate that a system is:

- reliable,
- meets its specifications, and
- meets the requirements of the user.

Careful and thorough testing and evaluation can improve system security by uncovering errors, omissions and other flaws in the application system's design and code. However, even careful and thorough testing and evaluation cannot assure that all flaws in the system will be found. Security requirements are usually stated in terms of events that must not occur under any circumstances. Testing is less effective in evaluating these negative requirements than it is in verifying the positive performance of the system under typical circumstances. Formal proofs of correctness are able to evaluate whether very general and even negative requirements are fulfilled; however, formal proofs are too difficult for widespread use at this time.

If security has not been considered throughout the development life cycle and if attempts to avoid flaws have not been made from the beginning, then the elimination of flaws identified by test and evaluation cannot be expected to improve security significantly.

6.4.1 Test Plan. A test plan for security should describe what is to be tested and what testing methods or tools will be needed. It should include tests that identify the system's response to abnormal, unusual, improbable, and illegal circumstances that may exist during both data input and processing.

The importance of testing for exceptional conditions is illustrated by the case of an accounts payable system where the software security control was that for every payment made, a record of the payment was sent to the appropriate regional office of the organization. The regional office was responsible for checking the payments manually. In this case, an appropriate administrative control was thought to exist and, thus, software responsibility was minimal. However, it was learned that a request for payment with a nonvalid (null) regional office identifier was still processed and a check issued.

The documentation of requirements, if it has been kept up to date, provides the basis for formulating the test plan.

6.4.2 [redacted] These techniques, which involve examination and analysis of the systems documentation and code, represent the most effective way to detect deliberate traps or other unauthorized modifications. However, due to the complexity of most systems and the limitations of automated techniques and tools [HANTS 76], it is not currently practical to analyze systems completely using static evaluation methods. In addition, static evaluation does not examine the system in a "live" or operational mode so that errors in the execution environment are not detected. Specific techniques and tools include:

[redacted] -Portions of the source code are evaluated to determine if they implement the design specifications and are free from errors. In most cases, it will be impossible to review all of the system's code. Generally, samples of the code will be reviewed—especially critical modules or critical portions of the code. The code review can be done by an internal test and evaluation team that is involved in the system development or by an independent (third party) team—either internal or external to the organization. Code review differs from peer review, discussed in section 6.3.1, in that the code review is performed by individuals who were not involved with the design and programming of the application.

[redacted] A few individuals can be challenged to find unknown weaknesses in the security controls. If a penetration study [LINDR 75], [LINDT 76B], [HOFFL 77], [HSIAD 79] is to produce useful results, the context within which the penetrators are to work must be clearly defined. Penetration studies to identify programming errors can be expensive and are useful only if someone believes that there are no remaining errors that can affect security. The failure of a penetration study does not demonstrate that the application is secure nor does success mean that the application is totally insecure in any practical sense.

[redacted] These software tools [LYONG 74] aid the evaluation process by providing details about specific characteristics of the source programs. Examples include:

- cross reference listings are an aid during code review and may be useful to identify "suspicious" variables and source statement references.
- the variables which can influence control flow decisions can all be identified or variables which could be read before any value has been assigned to them can be identified.

6.4.3 [redacted] These techniques involve executing the application system, or portions of the system, with test data and comparing the actual results with expected or known results.

Dynamic testing may not detect unauthorized modifications or insertions to the system's code since they can usually be concealed by a clever programmer. However, program analyzers (see below) can provide some support in this area.

During dynamic testing, production data files should not be used as test data. In addition, all other important data files should be taken off-line or otherwise protected during the testing period to eliminate the risk of their being destroyed or their contents inadvertently revealed.

Dynamic testing is only practical for selected test cases. Fundamental questions such as, "Which test cases should be chosen?" and "How many test cases are enough?" must be answered. The answers to these questions depend upon the system being tested and upon the experience of the testing team.

The following tools can be used to aid the dynamic testing process:

[redacted] A program analyzer is a computer program that collects data about another program's operation while that program is executing. For example, an analyzer can determine whether test inputs have caused all of a program's statements to be executed, or it can maintain a record of the number of times each statement of a program was executed. Program analyzers can be particularly useful for evaluating how thoroughly the test data has exercised the program being tested. In addition, they can be used to identify extraneous code that might be an unauthorized insertion.

[redacted] Security flaws can be hypothesized [LINDR 75], [LINDT 76B], [HSIAD 79] based on analogous flaws found in other systems and then tested for existence on this system. This is an effective approach for selecting test cases that are likely to find security flaws.

7. PRESERVING SECURITY DURING OPERATIONS

Following development and successful acceptance testing, the application will enter the operations phase (i.e., its production environment). Even though the primary controls that were imbedded in the application will begin to function and provide the expected protection, security measures should not be relaxed during this phase. There are a number of procedures and practices which can be used both to supplement and to complement software security controls [IBMCO 76]. These include protection and control of data during input preparation, off-line storage and dissemination of output; employee hiring, job assignment, termination and security training; responding to security variances; software modification; hardware maintenance and advance planning to ensure continuity of critical ADP services at all times.

7.1 Control of Data

Protection, control and integrity of data is a continuing necessity throughout the stages data passes through in an automated application. Primary controls such as data validation and authorization that protect on-line data were discussed in section 3. This section deals with the controls needed during the various stages of input preparation, data media storage and output dissemination.

7.1.1 Input Verification. Input verification is the process of assuring accurate transcription of source data to machine-readable form. In most cases, the transcription process is accomplished by manually keying the source data at an input device such as a keypunch machine or a terminal with a CRT display [JANCE 74], [MARTJ 73], [RCBCC 78C].

PROBLEMS ADDRESSED:

If not detected, errors made during source data transcription can be propagated through the data base by additional processing of that data. The resulting loss of data integrity can seriously affect agency operations or management decisions that use this data.

Processing erroneous data can cause unanticipated control flow paths to be taken in the application software leading to program halts or bringing the application system down.

Specific Techniques:

- Visual Verification. The keyed data is visually compared to source data. For visual comparison, the keyed data may be in the form of a punch card with printed characters recorded on it, a human-readable listing, or a display in a CRT. Since visual verification is time consuming and subject to human error, it is only practical when source data is entered on-line for immediate use and where input validation controls exist.

- Key Verification. The source data is keyed a second time and automatically compared with the transcribed source data previously keyed. Detected errors are then corrected, rekeyed and reverified. While key punch verifiers are the most common devices used for key verification, key-to-tape and key-to-disk devices can also be used for this purpose and have the additional advantage of recording the source data on a high-speed storage medium.

- Check Digit. A source data item contains an extra check digit which is derived from all the other characters of the item by a special algorithm. When the source data item is keyed, hardware in the key device calculates the check digit and compares the result with the check digit that was just keyed as part of the source data item. If they are the same, it is assumed that the keying operation was correct. In general, the check digit is not calculated when keying data but rather is part of the predetermined alphanumeric code for the item being keyed. Thus, check digits apply best to data items whose values range over a known set of values which are not likely to change frequently (e.g., part number, employee number, department number) [JANCE 74]. Check digits are particularly useful when entering data on-line or when verifying data read by optical character readers. They can also be used to validate data input (see sec. 3.1.2). Although this technique is effective in most cases, it is not infallible since it is possible to make compensating keying errors in a source data item which do not alter the check digit.

- Control Totals. To use this technique, the source data is divided into small groups or batches of the same type of work. Individual totals are monitored for important items (or all items if desired) in the group and these totals form an input source record which is keyed along with the group. After the keying process, the keyed items are retotaled and the results compared with the source record containing these totals. A discrepancy indicates a mistake in the keying operation or a mistake in calculating the original total. This technique will require processing capability to reevaluate and compare the control totals. In addition, there is a trade-off between the size of the group and the ease of locating the errors since the technique detects an error in a group but does not indicate in which record of the group the error occurred. In an extension of this technique, known as cross-totals, a group consists of a number of records, each containing the same type of items. If one thinks of the group as a matrix, specific (or all) items in the group can be totaled by row and by column and the row and column totals keyed with the rest of the group. After the keying process, new row and column totals are calculated, and compared with the originals. Unless compensating errors have been made, keying mistakes will cause some of the new row and column totals to differ from the original row and column totals. For each pair of new row and column totals where either one or both of the row or column totals differs from the original, the data item at the intersection of the corresponding row and column will be in error.

- Machine Readable Source. Devices which read source data directly, such as optical character readers [FIPSP 76C], [RCBCG 78C], eliminate the time and expense of key transcription and verification. Used in conjunction with check digits, this technique provides a powerful means of verifying the accuracy of source data entry.

Management Considerations:

Input verification is a costly and time-consuming process. It is only useful for detecting keying errors and even when the keyed data is also verified, there is still a possibility of error.

7.1.2 Data Storage Management. Data storage management [IBMCO 76], [CARRJ 77], [BECKR 77] deals with the maintenance of and access to data storage media.

PROBLEMS ADDRESSED:

Data residing on storage media are subject to accidental and intentional modification, destruction, and unauthorized disclosure. An agency's mission could be disrupted if its critical master files or backup and recovery data were altered or destroyed.

Specific Techniques:

- Controlled access to storage area. Data storage media, when not in use, should be located and maintained in an area to which physical access can be controlled. The only person who should have access is the one having custody.
- User authorization. Media should be released only to individuals who have specific authorization for their use. Authorization for access to certain sensitive data should not entail access to all sensitive data.
- Accounting procedures. The movement of data storage media outside the off-line storage area should be accounted for at all times in a log which includes, at a minimum:
  - identifying information,
  - destination,
  - authorized signature, and
  - estimated time of return.
- Backup of sensitive data. The surest form of protection against destruction and alteration of data is the maintenance of additional copies at another location. Deciding which data to backup in this manner is a matter of management policy, but once that decision has been made, the creation and removal of backup data offsite should be handled on a routine schedule.
- Data encryption. (See sec. 3.6.2.)

7.1.3 Output Dissemination Control. Controlling the dissemination of the output of programs processing sensitive data helps to assure continued protection for the data after it leaves the computer.

PROBLEMS ADDRESSED:

If output is not carefully handled, but merely left to users to remove from the printer and separate, it is very easy for them accidentally to pick up more than they should. Even if printout is carefully separated, because of the similarity of most computer output, it could be confused. Unless there is some sort of accounting known to users, there could be a tendency to dispose of unwanted output or even to use like-appearing output for the wrong purpose.

It is much easier to steal information in hard copy form than out of a computer or from storage media. Therefore data is particularly vulnerable at the output dissemination stage. The following techniques provide some protection:

- Logging receipts. Limiting the number of copies, numbering each copy, and requiring an authorized signature for each copy is the only absolute way of assuring that copies of output go to those for whom they are intended.
- Distribution by mail. Distributing output through a mail delivery system is another, though less certain, way of assuring that copies go to those for whom they are intended.
- Labeling. Automatically marking each page of sensitive output with the appropriate sensitivity classification will provide a warning to someone who is not authorized to receive a copy.

7.2 Employment Practices

There are a number of practices, most as valid in manual processing environments as in automated ones, which can serve to lessen the vulnerability of a system to harmful acts perpetrated either accidentally or maliciously by employees [IBMCO 76], [CARRJ 77].

PROBLEMS ADDRESSED:

Employees who deal directly with data are in a position to reveal it to others as a curiosity or for personal gain or for revenge. They are in a position to modify data to the data subject's benefit or detriment. They may also accidentally make mistakes which can affect the subject of the data or the organization adversely. In addition, employees who deal with financial or other valuable data files are in a position to introduce information which will make either themselves or others beneficiaries of the system.

Specific areas of employment practices that affect the security of an application include employee hiring, assignment of responsibilities, and termination.

7.2.1 Hiring Procedures

- Investigation of Criminal Record. The existence of a criminal record may, or may not, indicate that a person is dishonest. The risk an organization is willing to take in this area will depend on the sensitivity of its operation and the duties to be assigned to the employee.
- Checking References. While poor references can provide ample justification for not hiring an individual, good references should not be the only criteria for employment. Characteristics, such as integrity, are not necessarily invariant over time. People who have been honest all of their lives can become dishonest if the conditions are right. With this understanding, employment, character and credit reference should be checked as thoroughly as possible. Employment references can be used to determine an individual's competence to perform, or to be trained to perform, the tasks required on the job. Character references can be used to determine such things as trustworthiness, reliability, and ability to get along with others. Credit references can be used to determine a person's financial habits which can, in turn, be an indication of maturity and willingness to accept responsibility for one's own actions. The temptation of working with a system which could produce financial benefits might be more than an individual in financial straits could resist. Such a person would be easy prey for external subversion. Credit references can also be an indication that people in financial difficulty will need to take a great deal of time away from the job, or will spend a lot of time on the job taking care of their own affairs.
- Security Clearances. The Federal Personnel Manual System FPM Letter 732-2 sets forth the requirements for security clearances for personnel associated with computer systems in the Federal Government. Complying with the requirements of the FPM Letter will result in observing the above hiring procedures.

7.2.2 Assignment Procedures

- Assignment in writing. All tasks assigned to an individual should be enumerated and specified in writing, and the employee should be asked to acknowledge understanding of the assignment in writing.

- Separation of duties. The precedent for separation of duties comes from the pre-automation era, but that is not an excuse for abandoning this well-established concept now that most operations are automated. It is still a very effective risk reduction technique. An example of such separation, originating in manual operations but continuing into automatic operations, is the segregation of the responsibility for preparing payment vouchers from the responsibility for approving payments. The responsibility for each of the functions listed below should, to the extent possible, be assigned to separate individuals. If such separation is not possible, an attempt should be made to separate closely related functions such as applications programming, program design review, program test and evaluation, and program maintenance. It is also important to define each function clearly so that there will be no overlap in responsibility from one function to another.

  - Data collection
  - Data preparation
  - Data entry
  - Data base administration
  - Authorization to use data for specific purposes
  - Custody of automated data
  - ADP operations
  - Systems programming
  - Applications programming
  - Program design review
  - Program test and evaluation
  - Program maintenance
  - Security of ADP facility and equipment, including remote terminals
  - Data communications
  - Output dissemination
  - Privacy requirements not covered above
  - Internal audit
  - Security coordination and planning

- Acting in concert. A requirement for having two or more individuals performing distinct parts of the same task, usually within a specified time limit, is a further refinement of the concept of separation of duties. It requires certain individuals, in separate locations or possessing separate keys or having separate means of access to the same resource, each following certain assigned procedures independently of the other individuals concerned, so that the culmination of their activity will be the desired event. For tasks that are susceptible to fraudulent or other unauthorized activity, the probability of such activity successfully occurring is reduced when collusion between employees is required.

- Enforced vacations. It is well known that vacations contribute to the health of employees by relieving the tensions and anxieties that typically develop from long periods of work unbroken by extended time off. Tense, anxious employees are more likely to have accidents or make errors and omissions in performing their duties. In addition, if all employees are forced to take vacations, there will be less possibility that one of them can set up a fraudulent scheme which depends on the employee's presence either for continuity or for secrecy. Even if the employee's presence is not necessary to the scheme, enforced vacations can be a deterrent to embezzlement because the perpetrator would always fear discovery during the employee's absence.

- Job rotation. Moving employees from one job to another at random intervals will have the same deterrent effect on fraud that a policy of enforced vacations will. There will be the additional benefit of having employees who can take each other's places in cases of illness or termination.

- Account limitation. Placing upper limits on the amount of funds which can be handled by an employee or billed to a project account and on the computer time that may be used by a project account will deter attempts by employees at embezzlement as well as their attempts to use computer time for their own use or to sell it as a commodity.

- Limitation of physical access. Physical access to a computer facility should be carefully controlled by barriers so that an individual who is there for only casual reasons such as submitting a job or picking up output cannot move freely around the area. In particular, programmers, who usually feel they have a right to be there, should be denied access to the computer room.

- Limitation of data authorization. Access to data should be granted with deliberation of each individual's need for such access, rather than according to rank or position or precedent. Group authorizations should be avoided except when there is adequate justification.
- Observation by supervisor. Neither the fact that an employee has a security clearance nor the fact that the employee is a long-time trusted employee should obviate the need for continuing observation by a supervisor who knows the employee well and who is able to observe signs of stress or aberrant behavior.

7.2.3 Termination Procedures. These practices should be observed upon the departure of any employee regardless of the conditions or mood of the termination. Under certain conditions, when the employee is likely to develop feelings of resentment, it is probably better to grant severance pay and terminate the employee's contact with the organization immediately. In other cases, when the parting of ways is on friendly terms, the authorization necessary for the employee to wind up the job can be continued in effect.

- Revoking all authorizations. All authorizations which have been granted to a departing employee (e.g., authorization to sign for sensitive data, authorization to access certain data files) should be revoked, a letter to that effect should be written and the employee's name should be removed from all lists of such authorizations. If the departing employee's authorizations included the granting of authorizations to others, these should also be reviewed. In most cases, there will be no need to revoke such authorizations, but some agency's regulations may require that such authorizations be granted under the signature of the responsible individual. However, it is possible that an authorization review may uncover unnecessary or erroneous authorizations due to changes in job functions of employees without corresponding corrections to the authorization data.
- Retrieving all keys, badges, etc. All keys, badges and other devices which are used to gain access to premises or information or equipment should be retrieved from a departing employee.
- Changing locks. The combinations of all locks known to a departing employee should be changed immediately. Periodic changes, which should also be performed, are not sufficient.
- Deleting passwords. Not only should departing employees' passwords be revoked, they should also be removed from the list of effective passwords so that the deleted password cannot be reused.
- Reviewing employee's obligations to the organization. If there are special conditions to the employment, such as denial of right to use certain information, these should be reviewed with the departing employee and a letter to that effect should be acknowledged by the employee in writing.

7.3 Security Training

Security training refers to the training that is necessary to indoctrinate personnel in the performance of assigned security tasks and to security awareness training [FIPSP 74], which addresses the possible threats which can occur, what to do about them, and how to prevent them. While such training might seem to deal with the obvious and self-evident, every organization which has instituted such training reports very favorable results.

PROBLEMS ADDRESSED:

When employees are not informed of management's interest in security, they are not likely to take steps to prevent the occurrence of violations.

Employees cannot be expected to assume responsibility for assuring security unless they are trained in the steps to take and encouraged to do so.

7.3.1 Task Training. Task training in specific security responsibilities should include all the skills and detailed outlines of the steps to take to adequately perform security related duties, such as operating fire extinguishers or notifying management or restarting the computer after a power failure.

7.3.2 Security Awareness Training. This should be a part of the indoctrination for all new employees whether or not they also are assigned specific security responsibilities. It should also be a continuing effort, addressing new subjects as the state of the art progresses and including periodic reviews of the need for security. Initial topics should include management's emphasis on security, universal responsibility of all employees, data sensitivity (privacy, confidentiality, etc.), and the effect of delays in operations on the agency's mission. Later sessions could include such things as physical security, variance reporting and response, and contingency planning but should never become so technical that the audience will be lost.

BEST COPY AVAILABLE

7.4 Responses to Security Variances

The possibilities for security variances to occur in the operations area are multitudinous and, therefore, very difficult to describe. They can range from an unauthorized individual attempting to get into the computer room, to a fire extinguisher that does not work, to a door that has been propped open for convenience, to faulty water sprinkler heads. Anything in conflict with existing rules or anything that cannot be explained or anything that is otherwise unusual should not be ignored but should be reported immediately to a supervisor or other designated individual to minimize risks to the organization and to the safety and well-being of the employees.

There should be a means of reporting such situations in confidence and without the stigma of "whistle-blowing." The procedure should be simple and should not encumber the reporting individual or tie up the employee's time in the follow-up. The need for such reporting should be stressed in awareness training and examples presented of what could happen if certain occurrences passed without notice, such as failure to check the credentials of an employee attempting to enter a restricted area merely because the checking individual is familiar with the employee and knows that the employee has had the appropriate authorization in the past. If the procedures for variance reporting are convenient and employees are continually encouraged to make use of them, the follow-up of incidents can be revealing to management and lead to the establishment of workable procedures for correcting potentially serious security situations. The emphasis in reporting variances should be on the circumstances rather than on the people who take advantage of them. In spite of management's encouragement to employees to report security variances, management should not expect employees to cooperate completely, particularly when such cooperation may require employees to report on the activities of their friends and co-workers.

7.5 Software Modification and Hardware Maintenance

Software modification and hardware maintenance [CARRJ 77] are necessary to the continuing usefulness of any system, but it should be recognized that performance of these activities increases the risk that errors, accidents, and intentional acts can occur.

7.5.1 Modification of Software

PROBLEMS ADDRESSED:

While changes or additions to code are being made, errors could be admitted to a program which could destroy data, or modify it to the extent that it would not be usable, or code could be inserted to perpetrate fraudulent activity.

The same programming practices for security described in section 6.3 above apply when modifying software in an applications system. In addition, observing the following practices will provide additional protection for the system while making the changes.

- **Code reconciliation.** New code to be inserted should be compared to code being replaced and to specifications for the change.
- **Availability of old code.** The code which is being replaced should be kept readily available in the program library for a reasonable period of time (at least 4–6 weeks) in case trouble develops in using the new code.
- **Documentation.** All changes should be thoroughly documented, including the intended result, the reason for ordering the change, any special effect on data or output and, especially, any change produced in configuration.
- **Use of test data.** All changes should be subjected to testing for predicted and desired results but production data should not be used for this. Special test data which simulates production data should be maintained and updated for testing purposes. A copy of production data can be used as test data if it does not contain sensitive information.
- **Use of original tests.** The modified software should be subjected to the same testing as the original software. The tests described in section 6.4 should be repeated on the modified software and the results compared.

7.5.2 Maintenance of Hardware

PROBLEMS ADDRESSED:

During the time hardware is being maintained, any data which is on-line could be inadvertently destroyed or sensitive data could be revealed to maintenance personnel.

Specific Techniques:

- Removing live data. All production data should be taken off-line for the duration of maintenance.
- Clearing memory. The main memory should be zeroed to remove all traces of data and applications software.
- Regular Preventive Maintenance. Preventive maintenance should be performed on a regularly scheduled basis rather than on an "as needed" basis in order to prevent the kind of errors which stem from poor maintenance.

7.6 Contingency Planning

Contingency planning [CANNR 72], [FIPSP 74], [IBMCO 76], [CARRJ 77], [BECKR 77], [MARTJ 73], [SCOML 75] consists of the advance plans and arrangements which are necessary to ensure continuity of the critical functions of an application system. The plans should cover all events of total or partial cessation of operations or destruction of the data base or physical facility. Such planning should include procedures and availability of equipment for both automated and manual procedures.

OMB Circular A-71, Transmittal Memorandum Number 1, July 27, 1978, requires all Federal agencies to develop and maintain "appropriate contingency plans." Viewed from one perspective, contingency planning is a valuable computer security safeguard which can be selected as a result of a risk analysis to satisfy some of the agency's protective needs. In fact, a risk analysis should be the basis upon which an agency determines the levels and extent of contingency planning "appropriate" to its situation. Consequently, the following discussion of the kinds of planning relevant to application system emergencies assumes that such planning will be initiated following the performance of a risk analysis of the specific application system.

PROBLEMS ADDRESSED:

An agency can be prevented from carrying out its mission in a timely manner due to the inability to process data or to recover information vital to management decisions.

The two situations to which application systems are generally vulnerable are loss of processing availability and damage to data. It follows that the kinds of contingency planning required are:

1. what to do if the network or computing system serving the application is down, and
2. what to do if the data base becomes so fraught with errors as to preclude further use without compounding the damage.

In both instances, the criticality of the mission supported and the urgency of the requirement for system output must be considered. The less delay a system can countenance, the more extensive will the planning need to be. All plans must be carefully and extensively documented, kept current, and disseminated to all personnel who will be affected by them. Personnel must be trained in their assigned tasks and plans must be tested for feasibility.

The following kinds of planning are relevant for application system emergencies:

7.6.1 Identification of Critical Functions. The functions in an applications system which are critical to an agency's mission should be identified by performing a risk analysis, which will indicate not only which functions are immediately critical but also the time periods after which other functions may also become critical.

7.6.2 Alternate Site Operations. Another organization should be found which uses the same or compatible equipment and, which, during an emergency, would be able to allot time for critical functions. Such an arrangement will usually be reciprocal. It should be effected on a formal basis between the organizations involved, not left to a friendly arrangement between individuals. Planning should extend to ensuring the readiness at the alternate site of up to date, tested software and sufficient preprinted forms and any other special supplies (envelopes, stamps, binders, display devices, storage facilities, etc.) to last through one reorder cycle. In addition, a method must be devised for communicating with the alternate site (telephone lines, messenger, etc.). Adequate transportation facilities may have to be provided and specific personnel must be designated and trained.

[redacted] It may be possible that processing of critical functions can revert to the procedures used prior to automation of the application. Certain types of data are amenable to manual processing for short periods of time; for others, only a severely streamlined version of the function can be accomplished. When continuity is important, any special equipment required for manual operation must be close at hand. Necessary particulars include working space, availability of timely data, page copies of source data, conversion of data to manual use, special equipment, preprinted forms, communications, transportation, selection and training of personnel, and a process for recording all manual transactions.

[redacted] The easiest way to ensure against loss of data is to keep additional copies in readily accessible and/or secure places. Decisions concerning how many copies to keep, where to store them, and how long to keep them, will depend on the criticality of the application, the vulnerability of the data, and the size of the data bases.

7.6.5 Recovery of Data. Sometimes data which has been either accidentally or intentionally modified will be so altered that further transactions will only compound the damage. In order to meet such an eventuality, a list of key personnel for each data base should be prepared so that they can be summoned without delay when needed. The individuals concerned would probably be senior systems programmers and senior applications programmers. In addition, the list should include the security officer or someone else who could authorize the substitution of extraordinary procedures if necessary, such as access to the computer room if not ordinarily granted, access to computer resources not ordinarily authorized and use of unapproved programs on live data. All personnel who might be involved in such a situation, including the computer operator, should be instructed in emergency measures in advance. Lists of such personnel should be kept up to date and should contain names of alternates as well. A priority, based on experience, should be established for checking the source of the errors and a small data base should be maintained for testing recovery procedures.

7.6.6 Restoration of the Facility. If a computer facility is destroyed or damaged, there must be not only plans for continued operation, but also plans for the restoration or relocation of the facility. Recovery from destruction can include locating space or real estate; building or remodeling existing buildings; procuring hardware, supplies, office furnishings, and ancillary equipment; and transferring operations from the emergency site to the permanent one. Such plans will normally be the direct responsibility of facility management but applications systems management may well be called on for support especially in the areas of workload planning, space requirements, equipment requirements and moving operations. Advance preparation and periodic updating of the information necessary to these activities will ensure its completeness and timeliness.

Management Considerations:

- Even if contingency plans are documented, kept up to date, disseminated, and tested, and personnel are well trained, there may still be unanticipated or unfamiliar situations that require an emergency response in a time frame that is difficult to respond to adequately. Recognizing this, management should require that personnel who are responsible for security in general and contingency planning in particular maintain an awareness of accidents, disasters, emergencies, etc. that have afflicted automated systems in other Federal agencies or private organizations in order to minimize the possibility of similar events occurring at their facilities. The sprinkler system accident which occurred at the Bureau of the Census in 1979 is a case in point. Line personnel responsible for reacting to emergency situations should be briefed on those experiences that are relevant and discussions should be held to ascertain how similar incidents could be handled, should they occur. At a minimum, personnel should be constantly aware of the location and purpose of all emergency control switches.
- Management should also recognize that in some situations personnel will not have time to stop and read the procedures. Even well-trained personnel who have been active in periodic testing of contingency planning procedures cannot be expected to memorize all of them. Although it is obvious that more frequent testing will improve employees' familiarity with the plan and their readiness to handle emergency situations, it is also evident that additional costs will be incurred. One of the major cost benefit considerations in contingency planning is balancing adequacy of testing against the cost.

Cancel/Mar

FIPS PUB 87

FEDERAL INFORMATION
PROCESSING STANDARDS PUBLICATION

1981 MARCH 27

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

GUIDELINES FOR ADP CONTINGENCY PLANNING

CATEGORY: ADP OPERATIONS
SUBCATEGORY: COMPUTER SECURITY

1. INTRODUCTION

The continued growth in dependence on computers and on the data processed by them has increased the importance of plans to prevent loss of their availability. Only a few years ago it was reasonable to consider recourse to manual operations when automatic data processing (ADP) systems became unavailable. Today, there are but few situations in which it is even possible to revert to manual processes. Thus, contingency plans are necessary to minimize the damage caused by unexpected and undesirable occurrences (contingencies) in and about the ADP facilities.

Security measures are employed to prevent or detect accidental or intentional disclosure, modification, or destruction of data or loss of the means of processing those data. Contingency plans, on the other hand, should be designed to reduce to an acceptable level the consequences of any loss of ADP resources or capability; they are not just planned responses to major catastrophes. As stated earlier, the purpose of a contingency plan is to mitigate the damaging consequences of unexpected and undesirable events *of whatever magnitude*. A contingency plan must not be directed exclusively at reaction to catastrophically destructive occurrences. While it is clearly true that those who are responsible for ADP resources must plan for the possibility of such catastrophic loss, they must also plan against less-than-cataclysmic events which also seriously impede provision of data processing functions.

The probability of the occurrence of an undesirable event is generally inversely related to its magnitude. Usually, the greater the catastrophe, the lower the probability that it will happen. In other words, data processing operations are disrupted with far higher frequency by small problems than by large ones.

There is another relationship which, while not obvious, is quite important to the quality of contingency plans. The size or scope of a catastrophe and of its effect on data processing operations are often not directly related. In the absence of a good plan, minor damage can cause major problems. Conversely, with a good plan, even major damage may not result in serious losses.

The following pages describe an orderly process for the generation of contingency plans. This document, however, is not intended to be an all inclusive source of information on contingency planning. Other sources such as FIPS Publication 31, Guidelines for Automatic Data Processing Physical Security and Risk Management [20], and others listed in section 5 should be used as appropriate to obtain additional data and ideas which may be applied when preparing the plan for any particular facility. The procedure in this document is provided to save those charged with the preparation of such plans the task of discovering or devising yet another process for doing so.

While this document is addressed primarily to data processing management, the information in it is also relevant to those organizations which do not operate a data processing center, but obtain support from other agencies or service bureaus through interagency or contractual agreements. Agencies receiving this type of data processing support should not assume that the servicing activities have adequately provided for any type of contingency. Thus, this document may be used to assist in establishing the requirements for adequate support during contingencies and in reviewing any applicable contingency plans for such organizations. Further, this document can also assist those users in a distributed data processing (DDP) network who operate or are responsible for equipment. Managers of such DDP activities should also consider contingency planning for their operation, and how the operation fits into the overall contingency plan for the entire network.

Few ADP operations and facilities are so similar in equipment configuration, applications, environment, personnel situation (particularly), and relative criticality of systems that a general-purpose contingency plan of broad applicability can be drawn and applied equally well to more than one facility. For this reason, no recommended plan is provided here. Instead, a guideline is provided for preparing a specific plan suitable to the peculiar needs of each ADP facility. However, the contingency plan for any data processing activity, regardless of its size or scope of operation should, as a mimimum, address the following three elements:

- **Emergency Response**—Emergency response procedures to cover the appropriate emergency response to a fire, flood, civil disorder, natural disaster, bomb threat, or any other incident or activity, to protect lives, limit damage, and minimize the impact on data processing operations.
- **Backup Operations**—Backup operations procedures to ensure that essential data processing operational tasks can be conducted after disruption to the primary data processing facility. (Arrangements should be made for a backup capability, including the needed files, programs, paper stocks and preprinted forms, etc., to operate the essential systems/functions in the event of a total failure.)
- **Recovery Actions**—Recovery actions procedures to facilitate the rapid restoration of a data processing facility following physical destruction, major damage, or loss of data.

To the extent possible, contingency plan documents should be brief so as to facilitate their usefulness and acceptance by the users. The plan should be tested on a recurring basis and modified as changes in the data processing facility workload dictate. Critical applications should be operated on the backup system regularly to ensure that it can properly process this workload. (See sec. 4, Testing.)

2. ROLE OF MANAGEMENT

ADP facilities generally provide a service to one or more functional areas of organizations of which they are a part. Occasionally, they provide data processing support to several organizations. Recognition that the ADP shop serves in a support role is essential to the proper conduct of many aspects of data processing management. It is no less important to the generation of realistic, cost-efficient contingency plans.

Because the ADP facility normally provides vital services to the organization, the senior management of each organization should realize the critical nature of that organization's dependence on contingency plans. These plans, if carefully prepared and executed, serve to keep within tolerable limits the consequences of losses or damage to ADP resources. Economic feasibility in contingency plans requires carefully derived decisions as to what organizational functions are deferrable and for how long. As described later, the costs of these deferrals should be established.

It is practically impossible for such decisions to be reached entirely within the ADP organization. The ADP management is not usually in a position to assess accurately the relative importance to the whole organization of work done by the respective supported areas. Further, the relative cost of continued support of each in the face of adversity may vary quite widely. Thus, cost of support under unusual conditions must be considered. For these reasons, it is not only appropriate but very important that the senior agency management provide direction and support for contingency planning to the end that the agency continue to provide essential services following disruption of the ADP facility. Senior management should do the following:

- Demonstrate a firm commitment to the ADP security program by promulgating objectives and including responsibilities to attain those objectives in job descriptions and in promotion plans, when appropriate.

- Direct the establishment of contingency plans which are based on the results of a comprehensive risk analysis.

- Direct the support of the planning process by all organizational units servicing and serviced by the ADP facility. In particular, identify those elements of the organization which are critically dependent upon the ADP facility. Of extreme importance is the assistance of supported activities in identifying those vital records and data maintained by the ADP function, i.e., those which are essential to the sustained continuation of supported activities following a disruption of service or destruction of the ADP facility. (Responsibility for preparation of the plan should be with the ADP facility.)

- Direct the initial and periodic later tests of the workability and costs associated with the plan. (See sec. 4, Testing.)

- Direct the periodic revision or update of the plan as a consequence of information derived from the tests, and as a result of changing dependence of the organization on ADP. Likewise, a complete review of the plans should be made upon addition of new applications systems, reaccomplishment of a risk analysis or a change in any of the critical dependencies.

2.1 Risk Analysis

The development of a contingency plan to minimize the damage resulting from losses or damages to the resources, or capability of an ADP facility is dependent for its success on recognition of the potential consequences of undesirable events against which the facility needs protection. The facility is an assemblage of many resources. Some particular subset of these is needed by the facility to provide data processing support. These resources include people, programs, data, data processing hardware, communications facilities, power, environmental control, the physical facility and access to it, and even paper forms.

All resources are not equally important. They are also not equally susceptible to harm. Therefore, the selection of safeguards, and the elements of a contingency plan, should be done with informed awareness of which system functions are supported by each resource element (devices, programs, data, etc.), of the susceptibility of each element to harm (accidental or intentional), and of the consequences of such harm. In short, cost-effective protection of a data processing facility is heavily dependent on:

- An awareness of the facility's relative dependence on each of its component parts,
- Knowing, at least, in an overall way, what the chances are that something undesired will happen to each component,
- A determination of the ramifications of undesired things happening so that, things can be done to minimize either the chances of their happening, the loss if they happen, or both.

The maximum allowable cost of any safeguard is limited by the size of the expected losses which will be mitigated by that safeguard. Any safeguard or combination of safeguards must not cost more than tolerating the problems to which the safeguards are addressed. Satisfaction of these criteria clearly requires a process which identifies the expected losses as a consequence of undesired things happening to resources. Such a process is called a risk analysis.

In addition to providing a basis for the selection and cost-justification of security measures, a risk analysis provides data on time as a factor in assessing the possible consequences of losses of security. Knowledge of the consequences of not being able to perform each system function for specific time intervals is essential to the creation of contingency plans which are adequately responsive to the needs of the supported organizations.

With very few exceptions, a large percentage of an ADP facility's workload is deferrable for significantly long periods of time before the deferral causes unacceptable hardship. On the other hand, there is usually a small percentage of the workload which must be run because its delay would cause intolerable disruption. It has proven very difficult to guess reliably and accurately into which category each data processing activity should fall. It is also very difficult to guess very accurately the maximum tolerable delay for the processing of each deferred activity. A properly conducted risk analysis yields these data, which can then be used to justify or reject contingency plan elements based on actual, quantitatively-expressed needs of the supported organizations for ADP services.

A suggested risk analysis procedure which provides the desired data is described in FIPS Publication 65, Guideline for Automatic Data Processing Risk Analysis [22].

2.2 Critical Dependencies

As described earlier, the prompt recovery of an ADP facility from a loss of capability is dependent upon the availability of a variety of resources. The specific resources required are a function of the nature of the problem which generated the need for recovery. Some of these resources are absolutely essential to reestablishment of operations and, as such, are critical dependencies which warrant special care to assure their continuing availability and early recognition of a loss of capability. These things on which there is a critical dependence are usually in two distinct categories, which are:

- Resources under the direct control of the ADP management.

Within this category special care must be placed on determining which data are needed for backup and recovery purposes. There is generally a certain amount of information which is absolutely vital to the organization, and which would have the first priority in any emergency situation. This information must be identified very clearly to facilitate its availability. There exists also in any ADP facility much remaining data which are extremely useful to an organization, and which if they had to be regenerated could be very costly. This type of data should also be properly categorized and maintained. During a long term backup operation, it is quite normal for management to expect continued availability of many of the day-to-day ADP products and services. Obviously, if backup copies of the information are not routinely prepared and maintained, it will not be possible to provide acceptable service during backup and recovery situations.

- Resources under the control of other persons.

The data processing management can and should take the steps necessary to assure the continuing availability of resources currently under its direct control. More difficult, but no less important, is the acquisition of firm commitment to the contingency plan of those resources under the control of persons outside of the data processing area. The external commitments of critical resources must be reviewed frequently to see that they have not been forgotten or otherwise neglected by the organizations making those commitments. Rehearsals or tests are the most satisfactory way of assuring the adequacy of most such commitments of critical resources.

After disruption of processing at the regular location, it is rarely logistically, technically or economically feasible, and even more rarely essential to continue all normal activity at the alternate location. The tasks performed by the facility are not all of equal importance. Further, the relative importance of many ADP systems may vary with time of day and day of week or month. A plan which attempts to provide the means to continue all processing without regard to relative importance will require expensive standby capabilities which must be frequently exercised to assure availability and compatibility with normal activities. This is very rarely feasible and usually unnecessary.

2.3 Management Review and Approval

The review and approval process for a contingency plan should be carefully established to satisfy several important objectives, as follows:

- Make senior management aware of any dependencies upon it for supportive action. Ensure that management realizes that during an emergency, there may be some services which will not be provided, or otherwise available.
- Obtain management agreement on the assumptions on which the plan is based, including the dependence on other organizations for assistance.
- Communicate to management the existence of a plan and obtain approval of the plan.
- Obtain formal concurrence of certain other organizations upon which there might be dependence.
- Through required reading and acknowledgment by signature (usually on a separate card), inform key employees of their respective roles in the various recovery scenarios.

3. THE ADP CONTINGENCY PLAN

A reasonable, systematic approach to contingency planning and documentation of the plan demands adherence to a carefully conceived structure. This structure is needed to:

- Assure that all important areas are addressed,
- Permit ease of reference to sections of immediate interest or concern, and
- Facilitate revision by minimizing the effect on the whole document of changes in limited areas of concern. Therefore, unless there is very solid justification for doing otherwise, the documented plan should be in loose-leaf form, highly sectionalized, with each page numbered and dated and with means provided to identify changes from the previous version of each page.

A contingency plan should consist of three parts which address two distinct, mutually exclusive sets of activity:

- Preparation Phase
  - Part One, Preliminary Planning
  - Part Two, Preparatory Actions

These two parts should cover those things which should be or have been done in anticipation of a loss to lessen the damage or assist recovery.

- Action Phase
  - Part Three, Action Plan

This part should cover those things which must be done after the fact to minimize the cost and disruption to the supported organizational functions.

Each part of the plan is essential to its overall workability and effectiveness; therefore, no part should be considered more important than another. There are differences, however, in the manner of their presentation.

Part One, Preliminary Planning, which is the basic driver of actions to take in the succeeding parts should be completed prior to beginning the actual preparation of the remaining parts of the plan.

Part Two, Preparatory Actions, describes specific preparation steps in a number of areas relevant to the facility and should be developed in as much detail as seems potentially beneficial. Such material can consist of "how to" instructions, and lists of information to the extent necessary. There will be time to read this material, to become educated in the problems and their potential solutions, to weigh alternatives and to select appropriate measures. An essential element of this part of the plan is unwavering insistence that *all persons on whom there is significant dependence during contingency operations be familiar with their potential respective roles,* i.e., when implementing Part Three of the plan. These persons are selected because they already know "how to." The plan must not be based on the assumption that the document describing it can be retrieved after the catastrophe by those with a role in recovery who will then read the plan to learn how to do what is needed of them. Except for supporting data from Part Two (such as lists of people, telephone numbers, addresses, and skills, of locations of things, of required resources to support key functions, and the like. i.e., things that are generally tabular in nature and difficult if not impossible to memorize), it should not be necessary to read the plan to initiate contingency operations.

Part Three, Action Plan, should consist of clearly stated actions which are to be taken upon the occurrence of an emergency. Part Three is divided into three sections: Emergency Response, Backup Operations, and Recovery Actions. Each of the three sections includes those things which are to be done in response to a set of problem scenarios. These problem scenarios are derived in a large part from information in the risk analysis process and from practical working experience. They must be representative of the reasonable anticipated problems. Immediately following each problem statement or scenario should be a description of what is to be done in each category described (not how). One scenario may require actions, and be listed, in one or more sections of this part, e.g., a bomb threat (which does not result in any damage), minor power outages, etc., may necessitate action only under Section One, Emergency Response. A sustained power outage would involve action under Section One and Section Two, Backup Operations. An incident causing serious damage to the facility would, most likely, require steps under Sections One, Two, and Three, Recovery Actions. Examples of typical scenarios and sections of the plan which might apply include:

- Sections One and Two. Fire or structural damage elsewhere in the building resulting in no loss of life has resulted in denial of access to the data processing facility for three days. Return to the ADP facility after that period is anticipated, but it might be slightly longer.

- Sections Two and Three. Destruction of the facility with loss of all personnel working at that time.

- Sections One, Two, and Three. Total communications failure.

- Sections One, Two, and Three. A hurricane, earthquake, tornado, or other natural disaster occurs which cripples local transport, power and communications but does little physical damage to the facility.

The scenarios mentioned above are not necessarily appropriate to any particular facility. The ones which are must be selected and be sufficiently large in number and breadth that they offer useful guidance in directing recovery in actual loss situations and in the performance of tests and rehearsals.

3.1 Plan Structure and Contents

A recommended structure for the ADP facility Contingency Plan is shown in the succeeding instructions, and includes suggested topics and information for each part and section of the plan. (An example Plan Outline is included as Appendix One.) While no one format can be totally appropriate for all situations and facilities, the plan structure as suggested here should be readily adaptable to virtually all ADP activities, regardless of the size or scope of operation. An essential point to remember while developing a contingency plan is that it should be viewed as a valuable, useful tool for the ADP facility and not as an additional burden which somehow must be borne.

3.2 Part One—Preliminary Planning

This part of the plan should describe the purpose, scope, assumptions, responsibilities, and overall strategy relative to the plan. Misconceptions concerning these concepts are quite common and must be clearly addressed to ensure that they are communicated to those who must effectively respond to a contingency by implementing the plan. This part should conclude with a section which provides for recording changes to the plan. Each section of Part One and recommended contents are described below.

3.2.1 Purpose

This section should describe the reason and objective for having a contingency plan. If, for example, the continued well being of the data processing facility's parent organization is heavily dependent upon the data processing facility, it would be entirely fitting in this section to convince the reader, i.e., user of the plan, that the plan is not prepared merely to fill some arbitrary square in a checklist, but that the plan documents actions which are mandatory, essential and which must be subjected to continual review and testing to assure their adequacy in case of a contingency. A lackadaisical approach to contingency planning, if inferred or somehow communicated to the reader in this section, will set the stage for a corresponding less-than-enthusiastic acceptance of the plan and its requirements by the employees. The statement of purpose should convey the notion that the plan is a dynamic, on-going activity which includes not just things which are done in anticipation of a problem; that is, things done before the fact to mitigate the damage and to ease recovery, but also what is to be done when problems occur.

3.2.2 Scope

This section should describe in concise terms the extent of coverage of the plan, e.g., "This plan is applicable to the data processing facility located in building 1234. It includes all functions, i.e., data processing and ancillary services, administrative functions, etc., associated with the Data Processing Division. It also includes off-site storage facilities located at (location)." It is especially important for large activities with multiple data centers, communications facilities, etc., to clearly state in this section whether the plan encompasses all facilities even though not co-located, or applies to just specific locations as mentioned. The existence and location of plans for other facilities, if applicable, could be referenced in this section to facilitate their retrieval when needed.

3.2.3 Assumptions

A contingency plan is based on several categories of assumptions. Most can be established only after a quantitative risk analysis. The whole list of assumptions for inclusion in the document cannot be completed until well along in the planning cycle. Included in the set of assumptions should be the following:

a. Nature of the Problem

— The general nature and range of events against which the plan is directed.

— Events not addressed by the plan which, because of their low probability, do not warrant consideration in the plan.

— Events which are so extensive in scope as to negate the feasibility for early recovery of data processing operations.

— Events too minor in scope to warrant reflection in the plan. These are generally sufficiently frequent as to be considered a normal part of the operation and which are now accommodated routinely.

b. Priorities

Senior organization and ADP management have a critical need to understand the manner in which priorities are determined. The data sources, the extent of user agreement on the selected priorities, the risk analysis methodology, and other related matters should be described in detail adequate to a full understanding of the relative priorities to be observed in recovery of operations and of the rationale used to establish those priorities. In many organizations the relative criticality of the supported functions will vary with time of day, day of week and of month. Where appropriate, the description of priorities should reflect that situation.

c. Commitments to or Assumptions of Support

Recovery from any but minor, and relatively frequent, problems usually requires assistance in some form from groups beyond the immediate control of the ADP management. The assumption of such support, including letters of formal commitment by other organizations, difficulties in getting commitments, and related matters should be addressed. The list of the assumptions relative to resources might include the following:

— Availability of replacement hardware and licensed software.

— Availability of supplies possibly influenced by transportation problems in the event of a major problem.

— Utilization of another ADP facility and its formal commitment of support.

— Availability of people of all categories. (The mobility of employees after a natural disaster is frequently overestimated, particularly when that mobility requires leaving dependents in less than desirable circumstances.)

— Response of public utilities, particularly if there is a natural disaster of some kind.

— Availability of funds, including indication of gross amounts and possible sources.

3.2.4 Responsibilities

This section should document specific responsibilities as assigned by management to all activities and personnel associated with the plan, e.g., who within the data processing activity prepares the plan (by part, if appropriate), and who executes the plan. Whenever there may be a possibility of persons not knowing for what they are responsible when execution of the plan is required, the resonsibilities should be clearly specified in this section. When situations which require use of a contingency

BEST COPY AVAILABLE

plan occur, it is seldom during a period when the full complement of management is available. Therefore, it is critical that this section clearly delineate how the chain of command is to function when an emergency strikes. The document should explicitly indicate what emergency responsibilities are assigned and delegate the necessary authority to enable the selected individuals to carry out the assigned duties. Generally, for the emergency chain of command, it will be necessary to do this by function, e.g., shift supervisor, senior operator, rather than to named individuals. If the emergency chain of command duties are assigned in the plan to specific persons by name, and they are not present in the facility when needed, there is a distinct possibility that the entire operation may fail due to confusion about who is in charge.

3.2.5 Strategy

With relatively few exceptions, the selection of appropriate strategies should follow the risk analysis. Until the risk analysis is done, it is usually difficult to know the critical systems which must be maintained and the demands for resources which will be made to support those critical systems. Thus, it is expected that the strategy can be, at least tentatively, selected immediately after the risk analysis is complete. ADP and organizational management, as appropriate, should determine and have documented in this section the basic strategy to be followed by the ADP facility when implementing the Emergency Response, Backup Operations, and Recovery Actions (Part Three of the plan). This should be accomplished prior to planning any preparatory measures (Part Two of the plan). Information for use in developing strategy is categorized by area as follows:

a. Emergency Response

In the broadest sense, the strategy for this section is axiomatic, i.e., protect lives and property to the maximum extent possible. When developing a strategy to cover specific events, more complex actions and planning are necessary and must cover a very wide range of potential situations. To illustrate this, the strategy for coping with a severe hurricane will surely be different from that for a minor, easily controllable fire which creates smoke in the data center. In the first example, the strategy might include actions such as close the center, secure/transfer critical files, and release all personnel to allow them to assist their families. In the latter example, the strategy could be simply to execute power down procedures and evacuate all personnel to a nearby assembly point, e.g., across the street in fair weather, or to some pre-selected building in inclement weather. In any event, the strategies selected must provide a sufficient base upon which procedures can be devised which afford all personnel the immediate capability to effectively respond to emergency situations where life and property have been, or may be, threatened or harmed.

b. Backup Operations

Very few organizations will have ADP capability sufficiently dispersed geographically to permit the backup operations strategy to be unequivocally another site within the same organization. Most backup sites simply will not have sufficient equipment, personnel, supplies, etc., to sustain the complete operational requirements of another facility. The organizations which do not have redundant capabilities will be forced to develop a more detailed and difficult basic backup strategy. There are some ADP facilities which are now so highly centralized and so large that there is no other facility which can carry a considerable portion of the work representing the workload. For such activities, if it is necessary to continue providing data processing services after sustaining significant damage, then the strategy must reflect a plan to divide the facility between two or more physical locations selected to offer reasonable probability that enough capability will survive a major loss to provide means of processing the critical workload.

Contingency plans must not be limited to plans which provide for some processing time by another facility and the acquisition of secure, remote site storage of backup files. There are other categories of resources or capabilities beyond hardware and data which require careful consideration and which, in turn, must be reflected in the overall contingency plan, e.g., people, supplies, space, transportation. Strategies for backup of each of these other resources must be not only workable in maintaining availability of each resource through the full variety of threat scenarios, but they must be mutually supportive and compatible when employed in concert in the overall contingency plan.

Much of the planning for minimizing the damage caused by loss of, or damage to, one or more critically needed resources can be done without considering all of the other resources when the scope of the damage is relatively minor and does not result in crippling of the entire facility. However, when recovery is from a major event and must be accomplished at another location with heavy dependence on backup in most resource categories, the plans must be derived from a thoroughly comprehensive strategy for recovery from these catastrophic situations. Listed, as examples to consider, are several fundamental strategies for backup of an entire facility. Strategies for operation when less than total inoperability is the problem must also be developed. One single strategy, as in the case of loss of a facility, is rarely adequate because of the need to respond to a wide variety of problem scenarios.

- Strategy 1—No Hardware Backup

Some few organizations need an ADP facility to perform their mission, but will not be seriously harmed if they are completely without it for periods of time possibly as long as two weeks. It is the nature of these operations that they are rarely, if ever, dynamic, transaction-oriented, communications dependent shops. In these few cases in which dependence on ADP is not immediate and critical, it is not unreasonable to assume that the original hardware can be repaired or replaced at the current or another location in time to avoid major loss provided only that other dependencies, such as people, data, and programs, are suitably protected through backup procedures. *Believing that backup of hardware facilities is not required is not sufficient justification for ignoring contingency planning.* Further, a sound risk analysis must support the conclusion that no backup arrangement is required.

- Strategy 2—Mutual Aid Agreements

Mutual aid agreements are at least conceptually possible when one facility can accept, without serious harm to its supported organizations, the critical work of another temporarily inoperative facility. Technically practicable transportability of work between two facilities requires that data and programs from one be acceptable to the other without other than the most modest change and, preferably, no change at all. Rehearsals are essential, and it should be recognized that they are usually costly, and generate unwelcome disruption to the shop providing backup. The rehearsals must include full operability of the critical systems of the facility which is down. These practice sessions or rehearsals must be thoroughly realistic and not, for example, depend on the use of any resources from the inoperative facility for operation at the backup site. These are very difficult to conduct in a mutual-aid environment. To assure compatibility with the backup system, it is highly recommended that critical applications be run (daily, if necessary) at the backup facility as part of the normal job stream (with test data, files, etc.). Quite often, the site providing the backup support must drop some of its less than critical workload in order to provide the support to another facility. Also, the differences in security requirements between the sites must be considered. For example, clearance requirements at the backup site may preclude the entry of operators from the inoperative facility unless prior clearances have been obtained.

It is difficult at best to make mutual aid arrangements totally reliable. Changes in either system, (a highly likely occurrence) may instantly render the arrangement invalid. Further, management shifts may invalidate the arrangements with only short notice leaving a previously supported facility without backup.

While mutual aid agreements are conceptually feasible, they rarely, if ever, prove to be totally reliable. The penalty to the shop needing support of discovering in time of need that backup is not actually available is generally too great to warrant complete confidence in this strategy.

- Strategy 3—Contingency Centers

Contingency Centers are facilities established to provide a location into which an ADP organization which has lost its own facility can move temporarily to reestablish its operations, either completely or limited to critical systems only. These centers may be cooperatively owned by several organizations to back up the owners' facilities, or they may be established as profit-making ventures which sell rights to their use through membership fees, dues, and other charges. The evolution of these centers is still quite recent--too recent, in fact, for there to be a large body of experience to support their workability or to provide guidance as to the potential pitfalls to be avoided. Determining the feasibility of using such centers is not complex, does not seem to have hidden pitfalls, and thus should be relatively easy to do if based upon the results of the risk analysis. There are many situations in which such centers may well be the most cost-effective route to go, while there are others in which they are not an appropriate means of backup. Again, the decision must be made on a facility-by-facility basis.

Contingency Centers may be categorized as follows:

Empty Shells

Buildings with power, raised floor, and air conditioning but not data processing hardware. There are also buildings on which availability of floor space is maintained but in which there have been no preparations made such as the raised floor.

The empty shell provides a place to put replacement ADP hardware after a loss of the regular facility. A successful recovery in this contingency center environment requires consideration of a number of factors. These include:

— An adequate probability that *all* vendors of critically needed components can deliver soon enough to restore operations before unacceptable losses occur. This certainly implies several days before operation is restored.

Deliverable by vendor.

— An adequate plan can be drawn to get back from the shell to the proper permanent site with a level of disruption that is acceptable.

— The shell is sufficiently close to the permanent site to avoid severe personnel availability problems.

— There are reasonable arrangements for limiting the number of organizations which might be concurrently in need of the shell to preclude the possibility that service to some or all is inadequate.

Equipped Contingency Centers

Complete ADP facilities including communications capabilities. These are readily usable by organizations which have compatible hardware. These equipped centers are of two basic types:

— Those which normally operate as service bureaus but which plan to discontinue the provision of these normal, service bureau services in the event that a subscriber to the contingency center has a need for all or a portion of the backup center.

— Those which are not otherwise used except to rehearse contingency plans and to assure the operation of critical functions on the backup equipment.

The planned use of the equipped contingency center requires the prior consideration of several factors, including:

— Compatibility of hardware.

— Restoration of communications. This problem can be minor when dependence is on only a few lines, but it grows rapidly with increasing communications complexity.

— The cost of initial occupancy. If the cost of "declaring an emergency" is too great, the decision to use a contingency center will be unreasonably difficult, even when using such a facility would be very convenient to recover from less than a catastrophic event.

— Security considerations. This is particularly important if several organizations are sharing the same system.

— Availability of the facility for rehearsal. This is absolutely essential.

— Probability of too many subscribers needing the facility at the same time.

— Geography. If key people have to travel far, this is a major consideration. Subsequent recovery at home is also difficult if the key people are away at a remote site.

- Strategy 4—One Facility, More Than One Location

This is achieved by having ADP in two geographically separated locations, the smallest of which is large enough to carry the critical workload for the few days needed to reestablish the inoperative facility. This strategy does not imply the installation of excess capacity great enough to carry the critical work—only the physical dispersion of the normal capability into two or more locations. The economic feasibility of this is based on the frequently confirmed assumption that, for the majority of facilities, the critical workload is less than 50% (commonly less than 20%) of the total load so that no increase in total ADP capacity is required. Hardware often does not divide cleanly into two halves, but there is usually no requirement to have precisely 50% at each site. Any split which will suit the need for processing the critical work at either location is adequate, provided, of course, that the backup facility converts its workload to include only its critical functions.

Realization of all of the potential benefits of the two-location option requires that full capability to run critical workload exists at both locations. This generally requires availability of the full range of essential skills to be available at each site. This might, but does not necessarily, mean significant added costs. However, the feasibility of this depends heavily on the size of the operation being considered.

c. Recovery Actions

The strategy for recovery must be linked closely with that of Backup Operations as initiation of recovery actions may overlap, or be the next step after backup operations in restoring the ADP capability after partial or complete destruction of the facility, or other resources. The wide variety and scope of actions involved in recovery may dictate separating the specific recovery actions into two categories, i.e., short term and long term. Examples of common recovery strategies and scenarios are as follows:

- Repair/Restore Current Facility. ADP Facility Damaged—Backup Facilities Available for Critical Processing.

This situation which is fairly typical is heavily dependent on local conditions, e.g., how long non-critical workload can be deferred; continuing availability of equipment at the backup facility, etc. The short term strategy might be, simply, to defer non-critical work until the facility is restored, or, if possible accomplish manual processing. The long term strategy in this situation most likely would be to restore operations at the existing facility by using previously considered contractors and vendors for construction and other services needed.

- Rebuild Facility at Current Site. ADP Facility Destroyed, No Backup Facility/Hardware Available.

Hopefully, in this situation, management will have determined that there is no critical workload processed by the facility, and that short term recovery is not required, i.e., long term recovery consisting of acquiring equipment and a new facility will begin after occurrence of the disruption of service.

- Build New Facility at Different Location. ADP Facility Destroyed, Management Not Satisfied With Current Location.

If management has been actively considering the relocation of the ADP facility in the near future (1-3 years), the short term strategy may be to use a backup operations site. The long term strategy could consist of accelerating the preparation of the new facility at a different location. For this to be a viable strategy, actions for acquiring the new facility must already be well advanced.

When developing recovery strategy, careful consideration must be given to how DP equipment will be replaced and systems transitioned in time of need. If the quipment and applications are so unique to the organization that backup is not vailable, and replacement during an emergency is not a viable alternative, the ctivity must consider transitioning at least the critical systems before the fact to quipment which would be available during a crisis. The systems so transitioned hould be continually updated, tested and retained for use when needed. Thus, upon occurrence of a disaster, the organization will at least be capable of processing critical work at a backup site. With this availability of software to process critical systems, the recovery strategy then, may be to acquire equipment compatible with that used for backup purposes. (See sec. 3.3.4, Hardware.)

3.2.6 Record of Changes

An essential element of any volatile document, such as a contingency plan, is a method of preparing, posting and recording changes to the document. Entries in this section should include change number; date; pages changed, deleted, inserted; name of person posting change; when posted; plan distribution; and other information as local conditions warrant.

3.2.7 Security of the Plan

Once documented, the plan provides a significant amount of information about the organization which, if misused, could result in considerable damage or embarrassment. Consequently, the plan should be made available to just those personnel affected by the plan. Widespread or indiscriminate dissemination to persons outside the organization should be avoided. It is recommended that a specifically designated function, e.g., security officer, control the distribution to preclude any release of the information to unauthorized persons or activities.

3.3 Part Two—Preparatory Actions

If an organization has a properly designed and documented contingency plan, and there is a loss requiring its use, this section of the contingency plan will be a key part of the document to which reference is required to reestablish the data processing operation to normal. A primary point to remember is that after the fact, it is too late to prepare for the problem, and, as noted earlier, it is very important that all needed persons know the respective roles to be played in recovery without the need to acquire and study the document.

This part of the plan, if used as described here, will usually be critical to the emergency response, backup and recovery from all but the most routine problems. This part is also the most frequently changed section of the document, because it provides the lists of detailed information and procedures which are difficult to memorize. The number of sections needed, their size, and their content will vary with the nature of the ADP facility. Shops which support a wide variety of organizational functions, which have complex procedures involving numerous people, extensive off-site record storage, and heavy dependence on communications will always have more complex requirements for information to support recovery than will less complex, but not necessarily smaller facilities.

The sections which should be considered for inclusion in Part Two of the facility's contingency plan are shown below, along with definitive information about the contents of each of the sections.

3.3.1 People

No other functional element of the many which comprise an ADP facility even approaches the flexibility, adaptability, breadth of function, and versatility provided by the people who work in it or for it. No other element is so critical to graceful recovery from damaging losses. Unfortunately, the availability, while functioning in the desired and prescribed manner, of no other functional element is as difficult to factor realistically into a contingency plan as are people. Because of this, the workable plans are those which reflect, as the primary concern, this dependence on people and which accommodate the problems they present.

Replacement hardware, backup copies of data, duplicate programs, and new floor space will usually perform about as well as the originals without need for learning or motivation. People do not.

People can be expected to innovate, perform unfamiliar tasks, work under considerable stress and work long hours if they are in a reasonably familiar environment, particularly if they are not too deprived of the creature comforts found in their normal work environments. People tend not to perform complex tasks well in a physically stressful environment unless there are also strong motivations for doing so. In a recovery operation, heavy ego involvement, belief in the inherent importance of the organization's mission, intense loyalty to particular people in the management structure, or some combination of these provide the principal motivation for people to perform well. The planners of backup and recovery must assess for each facility the degree to which it is safe to place dependence on these factors.

care of dependents

When extreme weather conditions, such as floods, tornadoes, and hurricanes, have created the need to invoke a contingency plan, people with dependents of whatever nature (including families, houses, cars, airplanes, and boats) are often extremely reluctant to leave them exposed and not cared for to go to a geographically remote alternate site to effect backup operations or recovery of a data processing operation. It is under stressful conditions that they often find a variety of reasons for not going, or that they do not effectively perform due to unhappiness or concern for that which they left behind. This situation must be considered a major factor in the selection of a basic philosophy for alternate site operation. Without the necessary people, there can be no recovery.

If ADP operation is at two or more locations, under common management control, as described under Strategy 4, above, then the people at sites not suffering a loss, and backing up one which did, can be in a position to drop less essential, normal tasks and pick up the critically needed functions of the site which is not operational. These people at the still-functioning, undamaged sites are in familiar surroundings and best able to carry on the critical tasks without interjection of the array of personal problems possibly facing personnel in the site which is inoperative.

If two or more sites routinely provide backup for each other during periods of equipment changeover, failure, scheduled maintenance, and other interruptions and also for purposes of rehearsal of emergency recovery, then changeover to the alternate site is relatively easy. A potential management problem does exist, however, whenever the initial or "home" site also provides operators to the backup site. In this situation, the effect of the different supervisory and reporting structure must be considered, as there is a tendency for the operators assigned to the home site to dominate the two. This may cause personality conflicts as well as a degradation of operations for both units and may well negate the objective in establishing a second site.

It is assumed that the contingency plan will be formed about several problem scenarios ranging from disruptions caused by loss of local power (even with backup) or communications, fire elsewhere in the building resulting in denial of access to the ADP facility, to major equipment failures, to intentional damage by malcontents, or to destruction of the facility by whatever cause.

This section should provide names, addresses, and telephone numbers of all people who may be required in any backup or recovery scenario. (Any organizational policies about employee privacy should be observed when compiling, distributing and protecting this list.) Prior compilation of this information is essential, as it cannot be assumed that upon occurrence of an emergency available management personnel will be sufficiently knowledgeable of the individual skills, talents and experience of assigned persons to select those needed for a particular recovery situation. (See sec. 3.2.4, Responsibilities.) Thus, it will be necessary on any list to associate people, skills and management in recovery. Alternates for persons with peculiar skills or with skills in very short supply must be designated. Alternates should be selected, insofar as possible, from among those people not sharing a common exposure; that is, working the same shift or in the same physical area.

Every person recognized as important to the support of critically needed systems, as well as their alternates, must be aware of the dependence of the plan on them. (Critical systems should have primary and alternate personnel allocated to them.) These people must be informed of their recovery responsibilities and roles, and the roles should be rehearsed to the extent necessary. Sufficient additional training and experience must be provided to enough people to ensure that the skills necessary to recovery are available when needed.

There is absolutely no requirement to list people in order of relative importance to recovery. The list is awkward to build, difficult to use, and assumes, unreasonably, that the relative importance of these people is somehow independent of the nature of the problem.

There is a possible post-emergency problem which is, perhaps, impossible to factor into a contingency plan, but awareness of its potential is justified. It is this: people are often given far broader responsibilities during and immediately following an emergency than they normally have. Some people will perform far better than their peers and sometimes better than more senior people in the organization. In such situations it is sometimes quite difficult to get everyone back into the original and probably appropriate relationships. This potential problem should be borne in mind in considering personnel assignments in contingency plans.

3.3.2 Data

Data in any form are subject to a variety of vulnerabilities; precisely which are of significant consequence in any particular situation is a function of several factors. It is assumed here, again, that a quantitative risk analysis has preceded any effort to select plans to support backup and recovery through protection of data. It must be noted that the data protection needs of an effective, adequately comprehensive contingency plan are but rarely, if ever, satisfied solely by periodically putting copies of selected files in geographically remote vital records storage facilities.

It is very common that the total scope of a problem which creates a serious disruption of data processing is attributable solely to the accidental destruction of data for which there is no promptly available replacement copy. This stops processing until data can be recovered as certainly as a power failure would halt operations.

The dependence of many critical systems on prompt recovery of operations after data have been accidentally or intentionally modified or destroyed by unauthorized means demands greater availability than is possible if travel to and return from a vault several tens of miles away is required to regain access to those data. There are systems employed in the direct control of critical organizational activities, facilities, or operations so dynamic that even a few minutes loss of data can have serious consequences. These include air traffic control, complex facility management systems, command and control systems, airline reservation systems and others. The contingency plan must accommodate needs for prompt replacement of data.

Care must be exercised to make certain that multiple generations of backup files are taken so that the period spanned is short enough to satisfy the needs for currency (possibly in conjunction with journals of all file updates) and long enough to span the period needed for recovery. If the total period spanned by all stored copies of a file is too short, it is possible that all copies stored contain accidentally or intentionally incorporated flaws which require the availability of backup files. Thus, the storage of multiple generations is required but the periods spanned between retained versions are not necessarily equal. Again, if the rate of change is high and the dependency is great, it may be necessary to take file images as frequently as once per day or even more. However, if it is possible that a continuing flaw can induce a problem in the data and, if it can take as long as a week to become aware of the problem, then copies should be retained for longer than a week as well. Otherwise, all backup image tapes will be flawed before the problem is recognized, thereby making recovery very difficult.

It is essential that all data on which backup and recovery are dependent be adequately recorded, maintained in a current condition, and backup copies adequately secured. Almost every facility is driven by a fairly complex array of data, including not only those data in machine-readable form but also the data normally on paper in various offices, in memoranda, and even unwritten. To the extent that recovery of operations after a loss is dependent on these data, as identified through risk analysis, they must be copied and appropriately protected. (See sec. 2.2, Critical Dependencies.)

More advanced data processing facilities have adopted the Data Dictionary concept for accountability for data, programs, and related information [15,16]. There are many different implementations, but all which have come to common usage provide a very valuable means of managing data and programs. In fact, the data dictionary is a tool specifically for data processing management, in that it reduces the amount of duplicate data and facilitates the preparation of documentation.

If a data dictionary has been fully and properly implemented and copies of it maintained in a current and physically safe condition, it can provide much data which otherwise must be derived manually and written into sections of the contingency plan.

3.3.3 Software

In a sense, software, i.e., programs, are just a special case of data. They tend to be more stable than do data, but they are sufficiently subject to change that care must be exercised that fully current versions and all necessary supporting documentation are sufficiently protected against the threats postulated in support of the risk analysis.

Application programs induce relatively peculiar vulnerabilities into the operations of an organization, e.g., fraud potential. These vulnerabilities become significantly greater with increased program size and complexity. There are many justifications for directing close management attention to the desirability of adopting one or more of the formalized programming management processes, e.g., structured programming with mandatory review processes, etc. The avoidance of and recovery from losses is only one of the many reasons for going that route.

One of the many potential benefits of these program development disciplines is the lessened dependence on the programmers who initially wrote each program. Procedures which reduce module size and complexity and enforce documentation and programming standards serve to improve maintainability and lessen dependence on specific individuals. This also, then, serves to enhance greatly the ease with which these programs can be handed off to other ADP facilities, normally or in a recovery mode, without the need for the services of the program authors.

As with conventional data, the identification and utilization of programs when a contingency plan is exercised are assisted greatly by a properly established and maintained data dictionary as mentioned above. A data dictionary system readily depicts the relationships of programs to jobs, to data, to functional areas of supported organizations, and to people and more, as may be needed.

If normal operations are in any way impeded when any author programmer is not available, this should serve as adequate indication that improved programming management is perhaps essential to a workable contingency plan.

It is very important that there be formal agreement with vendors of licensed programs, the copying of which is forbidden, to maintain the ready availability of replacement copies within a specifically stated time period. The written agreements with the vendor(s) should be effected to allow use of the software as needed during contingencies and testing at the home site, as well as the backup location.

3.3.4 Hardware

It is frequently said that, of all of the resources on which an ADP facility has dependence, the data processing hardware is the most readily replaceable. This is only approximately correct; it is not uniformly so. Hardware of reasonably recent or current manufacture and produced in quantity usually will have a higher probability of availability to the facility needing it for replacement than will devices of which this is not true. Devices which offer potential for difficulty of replacement include these:

- Those which employ a complex array of optional features and, as such, are effectively customized. Complex communications controllers, as an example, have the potential for this problem.
- Those manufactured in small quantities, including specialized devices.
- Those which are application sensitive, such as check sorters.
- Those approaching or having reached obsolescence.
- Those manufactured by organizations no longer in existence.

The ease of replacement of hardware is usually a secondary consideration in planning recovery of critically needed systems. Relatively few organizations can wait for the period of time required for hardware replacement to bring up those systems needed earliest after a loss. Even readily available hardware requires a few days. However, some very important systems which can be delayed a few days may be heavily dependent on device replacement. Contingency plans should minimize, to the greatest feasible extent, this dependence on rapid replacement of hardware.

Operations split across two or more geographically-separated locations (Strategy 4) (see sec. 3.2.5b) so that the important systems can be run on one of two or more sites under common management will be relatively free of heavy dependence on rapid hardware replacement. Similarly, the equipped shell (Strategy 3) (see sec. 3.2.5b), if otherwise feasible and if it can provide all of the devices important to the operation, offers another option.

As with all other resource elements addressed in the contingency plan, the prompt replacement of hardware is heavily dependent on pre-loss planning. It is usually difficult for a vendor of hardware devices to make a definitive, long-term, binding commitment to replace a specific piece of equipment within a specific time period after a loss. Availability varies dramatically with the passage of time, particularly as it involves the currency of manufacture. Another major factor is the number of facilities employing the device in question and which are affected by the same source of loss. An area-wide problem might create far more severe demands on hardware vendors than does a loss limited to a single facility.

That portion of the contingency plan which involves the categorization of operations into those which will be conducted, after loss of the facility in which they are normally run on another, existing facility and those which will be deferred until the initial facility has been restored must reflect a realistic assessment of the problem of acquiring all hardware on which each such function has an absolute need. Although the hardware vendors have, for the reasons described, difficulty in making formal commitments on the availability of replacement gear, it is reasonable to expect of them candid, if informal, identification of problems in this area; problems which may not be readily apparent to contingency planners. This information should be solicited.

3.3.5 Communications

The size and complexity of a communications network supporting an ADP facility is a major factor in contingency planning. It is, however, of no greater importance than the relationship of that network to the time-critical systems. The existence of the network cannot be assumed to define a critical dependence on it—or on all of it. This dependence must be known before steps are taken to provide communications backup and recovery.

The restoration of communications, even after major disruption, is usually much quicker than the establishment of the same capability at another location. Unless a plan is in place and that plan is agreed to, including schedule, by all parties who will have a role in establishing communications at an alternate site, then recovery of communications at an alternate site within a reasonable period cannot be assumed.

There are facilities which, either fortuitously or by plan, are located so that it is feasible to have cables arriving at the facility from two different central offices of the telephone company. Even if all communications are normally with only one central office, if means are agreed to by the telephone company to route communications to the alternate central office in the event of the loss or disruption of the one normally used, significant protection against disruption of these communications will be gained. Some facilities have found it desirable and feasible to normally use leased or dial-up lines brought in through two central offices.

Restoration of lines to the original site does not accommodate the needs generated by the destruction or severe disabling of an entire facility. Under that circumstance, all communications directed to the initially-used site must be made available to the site at which the operations supported by them will be conducted.

There is generally available from common carriers a means to switch leased telephone lines under remote customer control from the initial termination to an alternate site. The switch, the control of the switch, the cost of the line to the switch and the costs of the lines to the initial location and to the alternate site are all tariffed separately. The switching of lines in this manner may be economically feasible if the number is small, the distances moderate and the dependencies great.

An alternative to the leased switch is to maintain the capability to route all communications to each of two sites, described above as Strategy 4 (see sec. 3.2.5b). With this alternate no physical changes are required and all communications supporting critical systems are directed to the remaining site.

The economic feasibility and the time dependencies must be examined carefully to determine which approach to communications backup is best for any specific facility.

3.3.6 Supplies

With the exception of a very few items which might be peculiar to a particular facility, most supplies are catalog items with reasonable availability. However, for most facilities there is a sufficiently large number and variety of such items as to make plans for stockpiling a modest quantity at another, safe location a necessary step. When every effort is being made to restore operations, even after only a relatively minor disruption, valuable time is easily lost in locating things of limited dollar value, such as tape cleaners, floor tile pullers, labels, and marking pens. This time can be saved by advance planning for the availability of such items.

Facilities which consume large quantities of paper should have available a buffer supply of a size adequate to maintain operations after a disruption of supply until a normal supply situation has been restored.

The contingency planner should bear in mind that the provision of supplies is just not a minor task to be undertaken as a part of recovery from some major disruptions such as fire, flood, civil unrest, hurricanes, tornadoes, or airplanes flying into the building. The whole scope of the catastrophe may be the loss of availability of printer paper through physical damage to a vendor's facility or anything else which results in a business interruption there or in the means of transport from there. If the critical jobs performed by a system result in an essential paper output and if paper supply is interrupted without adequate backup, then the effective use of the system is denied as surely as if physical access to the ADP shop is denied.

Particular care must be given to the continued availability of special forms on which there may be a critical dependence. The replacement lead time on these is often measured in weeks. An adequate buffer supply should be stored off-site in a location not generally susceptible to the problems reasonably anticipated at the normal storage location.

3.3.7 Transportation

Events which disrupt transportation of people or supplies might have serious disruptive effects on the ability of data processing facilities to operate effectively. Of greater importance here, however, is the effect of loss of transport on recovery.

Area-wide power failures almost uniformly cripple all urban transport, including automobiles. Earthquakes make roads impassable. Labor difficulties can seriously impede public transport. These situations generally argue, not for plans to provide alternate means of transport, but for consideration of transport as a determinant in selecting an alternate site for conduct of critical work. The location of the backup capability should be free of whatever external problems are hampering the supported facility. Just as the supporting site should be served by another power company and another communications carrier, it should also be served by other forms of transport or not be susceptible to damage from interruption of transport.

3.3.8 Space

The provision of space into which an ADP facility can be placed after loss of an original site can be considered for two purposes, as follows:

- Space which can be used temporarily while the original site is being rehabilitated.
- Space into which the ADP operation can relocate with relative permanence.

Relocation at a temporary site usually requires installation of hardware at the temporary location and then relocation to the permanent site. This, then, implies two physical moves of processing operations with the significant disruption usually attendant on such moves. The expense is not inconsequential, particularly as it usually involves full site preparation at both locations.

A move from a damaged site to prepared or unprepared floor space after loss of a facility cannot usually be done in less than 1 week, and might take several weeks if equipment must be acquired. It should not be considered as a means of recovery of support of critically needed functions. It should only be done to restore full operation.

Recovery on the same day as the loss of capability, or within 2 to 3 days, must be at a location which is already populated with virtually all of the devices, communications, power and environmental control needed to support the critical functions.

Plans for space into which the facility can be relocated should, whenever possible, reflect future growth plans. Ideally, any move should be into a location at which the facility can stay permanently and expand as may be needed.

To the greatest extent possible, contingency plans should be drawn to minimize the possibility of temporary moves which require the diversion of key people, funds, management attention, and devices from the earliest possible recovery at a permanent location. There is but little to be said for mounting a major effort to become operational in the wrong place.

3.3.9 Power and Environmental Controls

Uninterruptable Power Supply (UPS) Systems provide three potentially useful functions. They are:

- Protection against power line transients which can provoke a system interruption requiring a restart and with the potential for damage to data.
- Provide a short period, usually from 15 to 30 minutes, in which the system can be stopped gracefully following a loss of primary power from a public utility.
- Provide a short period following a primary power failure during which time a standby generator can be brought into operation to support the data processing operation or at least the critical functions.

There are installed systems employing UPS for which instrumentation indicates that the UPS prevents as many as several dozen system crashes per year. This is a meaningfully large number when applied to large, real-time systems. On other systems which have little real time or online activity or very clean power, the UPS may not be justified. The determination must be made on the basis of a careful analysis of each specific facility for which it may be considered.

It must be considered in contingency planning that an unanticipated move from a complex environment providing UPS, chilled water, and 440 cycle power to a new location is not sufficiently simple to consider delaying critical systems until the move is complete. Again, as has been stated repeatedly above, provisions must be made to process critical systems on other equipment until the move is complete unless the critical load is deferrable for as long as a week. It is possible under highly fortuitous circumstances to complete a move to a new site in less than a week, but heavy dependency upon it is indeed risky.

Preparation of a backup site complete with power and all environmental systems would greatly ease the difficulty of a move to a new site, but the cost must be carefully considered. Under some circumstances where there is a plan for future growth into a site, the early preparation of the site as a contingency site can be cost justified.

It is particularly advantageous to select a backup site which is outside the "local" power grid, as this will provide alternate site processing capability when widespread local power outages occur.

In the event of partial failure of environmental controls (e.g., the air conditioning system is cooling, but at reduced capacity), it may be possible to selectively power-down less needed equipment in order to keep the computer operational. To prepare for this eventuality, a list of equipment which might be temporarily taken out of service should be prepared and maintained.

3.3.10 Documentation

This section of the plan should describe all backup documentation which is kept in the off-site facility so as to facilitate its retrieval in case of need. When preparing this section, it should be noted that one of the most critical elements of an effective data processing operation, yet, also one of the most neglected, is documentation. Without clear, concise and complete documentation, all but the simplest operations will flounder, particularly when contingency operations are effective and personnel are performing additional tasks or duties for which they are not normally responsible. For this reason, a complete set of all pertinent documentation such as Computer Operation manuals, Users manuals, Program Maintenance manuals, etc., (see FIPS Publication 38 [21], Guidelines for Documentation of Computer Programs and Automated Data Systems) should be stored in a secure off-site facility. Likewise, copies of the Contingency Plan, including equipment inventories, alternate site agreements, etc. should be stored in a secure location which is sufficiently removed from the main facility so as not to be subject to the same major hazards, e.g., flood.

3.4 Part Three—Action Plan

This part of the plan should consist of the "what to" actions to be accomplished by those personnel or activities identified in Part One, Responsibilities. As previously indicated, and restated here for emphasis due to its importance, when an emergency strikes, the people must already know "how to" respond. Therefore, it is expected that this part of the plan will consist only of concise, short instructions of the specific actions to take as a response to each of the problem scenarios which were earlier developed (see sec. 3.2.5a-c, Strategy) for each of the three categories listed below.

3.4.1 Emergency Response

Include in this section the immediate actions to be taken in order to protect life and property and to minimize the impact of the emergency. It is recommended that a separate list of actions be developed and maintained for each of the problem scenarios. For example, different responses are required for each of the occurrences such as bomb threats, power outages, air conditioning failure, fire alarm, etc. To facilitate use of the listings, more than one copy of each will be required in addition to the master, or file copy. The number required is dependent on the size of the facility and the ease with which needed sections may be retrieved for use. In any event, it should never be necessary to search for a needed portion of the plan after an emergency requiring its use has occurred. Following is an abbreviated example of the actions which might be included as a response to a sudden power outage of unknown duration (assuming UPS is installed and operative):

- Initiate power down procedures
- Notify key personnel
- Notify customers of disruption of service

The detailed instructions on how to accomplish the tasks listed above, and others as applicable to the facility should be located in Part Two of the plan under the specific categories of Power, People, etc.

3.4.2 Backup Operations

In this section, describe what must be done to initiate and effect backup operations, separately for each of the scenarios developed. For example, if the scenario is: major power outage of expected 3-day duration; backup operations at primary alternate facility necessary, the list of actions to take might include, but not be limited to:

- Notify alternate facility
- Notify backup team
- Notify customers of disruption of service
- Arrange transportation
- Retrieve backup supplies
- Assemble copies of software, data, documentation, etc.

Any "how to" instructions for each of the above areas should have been included in Part Two of the plan under preparatory actions.

3.4.3 Recovery Actions

As in the two preceding sections, the instructions in this section should be limited to describing what to do in effecting recovery from the situations documented in the problem scenarios. For example, if the scenario is: the ADP facility has been damaged, some equipment destroyed, but critical applications may continue to be processed. The action list in this case might reflect items such as:

- Perform survey to determine specific facility, hardware or data damage and losses.
- Retrieve backup files, as necessary. (Consider the possibility that if the primary file is destroyed and if the only backup copy of the file is retrieved from the off-site storage, no further backup is available until the backup file is copied.)
- Submit equipment order to vendor. (Equipment data should be available from Part Two of the plan.)

4. TESTING

One of the more important aspects of successful contingency planning is the continual testing and evaluation of the plan itself. Quite simply, a plan which has not been tested cannot be assumed to work. Likewise, a plan documented, tested once and then filed away to await the day of need provides no more than a false sense of security. Data processing operations are, historically, volatile in nature, resulting in frequent changes to equipment, programs, documentation, customer requirements, and often even in the way daily business is conducted. These actions make it critical to consider the plan in the same context, i.e., one in which frequent changes occur. A plan quite adequate today may be woefully unsatisfactory 2 months, or less, from now. Suffice to say, that if the ADP contingency plan is not subjected to continual and rigorous management review as well as to in depth testing on a scheduled basis it will fail when needed.

4.1 Test Plans

The devising of test plans which adequately and reliably exercise the contingency plan themselves require considerable skill and great care so as to meet the objective of providing tests which are entirely realistic while still being economically feasible. Care must be taken to see that the tests involve the most important systems to be supported in the contingency environment. The testing of the simpler jobs may be desirable initially, but such tests do not provide adequate assurance that the critical jobs will run.

4.2 Conducting Testing

A good argument can be presented that the only method to test a contingency plan completely is to actually cease or otherwise disrupt operations at the facility for which the plan has been prepared; however, this is seldom practical, and quite possibly could, in itself, create actual losses in capability. It is generally only necessary to assume that operations at the home site are disrupted or otherwise not available. For example, it is not essential to have an actual fire in order to test the emergency evacuation procedures. What is needed is an understanding with the fire department and documentation of the specific test procedures to follow in simulating the fire and emergency condition. Likewise, to test backup operations at an alternate site, it is not mandatory to cease operations at the home site, but rather to gather copies of all needed data and other information required to actually begin operations at the alternate facility. In situations such as this, the test most heavily inconveniences the supporting (alternate) facility which is assumed to be unharmed in the simulated catastrophe, or disruption of service.

4.3 Test Plan Documentation

The test plans should form a formal part of the contingency plan documentation and be as fully subject to the review and approval process as the other sections of the plan.

5. REFERENCES AND ADDITIONAL READINGS

[1] Becker, Robert S., The Data Processing Security Game: Safeguarding Against the Real Dangers of Computer Abuse. Pergamon Press, Inc., Elmsford, NY (1977) ISBN: 0-08-021790-7.

[2] Broadbent, D., Contingency Planning, The National Computing Centre, Oxford Road, Manchester, United Kingdom (1979).

[3] Browne, Peter S., Security: Checklist for Computer Center Self Audits, AFIPS Press, Montvale, NJ 07645 (1979). (Order from AFIPS, 1815 N. Lynn Street, Suite 800, Arlington, VA 22209.)

[4] Canning, R., Computer Security: Backup and Recovery Methods. EDP Analyzer. 10(1); (1972 January).

[5] Carroll, John M., Computer Security, Security World Publishing Co., Inc., Los Angeles, CA (1977) ISBN: 0-913708-28-3.

[6] Computer Control & Audit, Institute of Internal Auditors, Altamonte Springs, FL (1978).

[7] Courtney, Robert H., Jr. and Orceyre, Michael J., Considerations in the Selection of Security Measures. Washington, DC: GPO; 1978 June; NBSSP 500-33. Available from: GPO, Washington, DC; SN 003-003-01946-1.

[8] Courtney, Robert H., Jr., Security Risk Assessment in Electronic Data Processing Systems, Technical Report 21.700, IBM Corporation, Kingston, NY 12401.

[9] Checklists and Guidelines for Reviewing Computer Security and Installations, Management Advisory Publications, P.O. Box 151, 44 Washington Street, Wellesly Hills, MA 02181 (1975).

[10] Data Security and Data Processing, Vol. 1-6, (G320-1371, 1372, 1373, 1374, 1375, 1376) IBM Corporation, White Plains, NY 10604 (1974).

[11] Data Security Controls and Procedures—A Philosophy for DP Installations, G320-5649-00, IBM Corporation, White Plains, NY 10604 (1976).

[12] Disaster Preparedness, Office of Emergency Preparedness Report to Congress, Stock Number 4102-0006, Government Printing Office, Washington, DC (1972).

[13] Hoyt, Douglas B., Ed., Computer Security Handbook, McMillan Information, New York (1973).

[14] International Business Machines Corporation (IBM). An Executive's Guide to Data Security—A Translation from an IBM Svenska AB Publication. IBM World Trade Corporation, New York, NY (1975 October).

[15] Institute for Computer Sciences and Technology, A Survey of Eleven Government-Developed Data Element Dictionary/Directory Systems. Washington, DC: GPO; 1977 August; NBSSP 500-16. Available from: GPO, Washington, DC; SN 003-003-01817-1.

[16] Leong-Hong, Belkis, and Marron, Beatrice, Technical Profile of Seven Data Element Dictionary/Directory Systems. Washington, DC: GPO; 1977 February; NBSSP 500-3. Available from: GPO, Washington, DC; SN 003-003-01725-6.

[17] Martin, James, Security, Accuracy and Privacy in Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, NJ (1973).

[18] Martincic, J. A., A Disaster Recovery Plan, Journal of System Management, 1976 February.

[19] National Bureau of Standards Building Science Series 46, Building Practices for Disaster Mitigation (1973 February).

[20] NBS. Guidelines for Automatic Data Processing Physical Security and Risk Management. Washington, DC: GPO; 1974 June; FIPS 31. Available from: NTIS, Springfield, VA; FIPS-PUB-31.

[21] NBS. Guidelines for Documentation of Computer Programs and Automated Data Systems. Washington, DC: GPO; 1976 February 15; FIPS 38. Available from: NTIS, Springfield, VA; FIPS-PUB-38.

[22] NBS. Guidelines for Automated Data Processing Risk Analysis. Washington, DC: GPO; 1979 August 1; FIPS 65. Available from: NTIS, Springfield, VA; FIPS-PUB-65.

[23] Nielson, N. R., Ruder, B., Madden, J. D. and Wong, P. J., Computer System Integrity, SRI International, Menlo Park, CA (1978).

[24] Parker, Donn B., Crime by Computer, Charles Scribner's Sons, NY (1976).

[25] Pritchard, J. A. T., Contingency Planning, The National Computing Centre (U.K.), Computer Security Services (1976).

[26] Report to the Congress of the United States, U.S. General Accounting Office, Most Federal Agencies Have Done Little for ADP Disasters (AFMD-81-16), 1980 December 18.

[27] Schabeck, T. A., Emergency Planning Guide for Data Processing Centers, Assets Protection Journal, 500 Sutter Street, Suite 503, San Francisco, CA 94102 (1979).

[28] Wong, K. K., Computer Security Risk Analysis and Control, Hayden Book Co., Inc., Rochelle Park, NJ (1977).

6. APPENDIX ONE

CONTINGENCY PLAN OUTLINE

Part One—Preliminary Planning

1.1 Purpose

- Reason for plan
- Objectives

1.2 Scope

- Applicability of plan
  - Data center 1
  - Data center 2

1.3 Assumptions

- Events included
- Events excluded
- Priorities
- Support commitments

1.4 Responsibilities

- Plan preparation/maintenance
- Emergency chain of command
- Operations supervisor
- Shift supervisor

1.5 Strategy

- Emergency response
- Backup operations
- Recovery

1.6 Record of Changes

- Change sheet
- Plan distribution

Part Two—Preparatory Actions

2.1 People

- Complete listing of assigned personnel with address, phone number, etc.
- Emergency notification roster(s)
- Team composition
  - Recovery Team A
  - Recovery Team B

2.2 Data

- On-site inventory
- Off-site inventory
  - How/when rotated
- Critical files needed for backup site processing

2.3 Software

- System
  - On-site inventory
  - Off-site inventory
    - How/when updated
- Applications
  - On-site inventory
  - Off-site inventory
    - How/when rotated

2.4 Hardware

- Inventory list reflecting vendor, name, address, etc.
- Emergency acquisition agreement
- Sample order forms, etc.

2.5 Communications

- Current on-site requirements
- Requirements for backup site(s)

2.6 Supplies

- List of critical supply items with all necessary information (e.g., stock numbers for ordering)
- List of vendors who provide supplies
- List/location of supplies needed for backup site processing

2.7 Transportation

- Requirements for recovery operations/backup site(s)
- Procedures for obtaining emergency transportation

2.8 Space

- Current site requirements (lay-out of facility)
- Backup site space available, by site

2.9 Power and Environment

- Current site requirements
- Backup site requirements

2.10 Documentation

- On-site inventory
- Off-site inventory
  - How/when updated
- List/location of critical documentation needed for backup site processing

2.11 Other

- Alternate site agreements
- Contracts

2.12 Test Plans

- Plan A
- Plan B

Part Three—Action Plan

3.1 Emergency Response

Scenario 1
Scenario 2
Scenario n

3.2 Backup Operations

Scenario 1
Scenario 2
Scenario n

3.3 Recovery Actions

Scenario 1
Scenario 2
Scenario n

NOTE: The exclusion of any item in the examples above does not imply that further entries may not be required for any facility. The purpose of the example entries is to suggest, generally, possible relevant entries for each facility's contingency plan. Most planners will undoubtedly discover that in order to provide complete coverage, further expansion of the outline will be necessary.

7. GLOSSARY

Backup Operation

A method of operation to complete essential tasks (as identified by the risk analysis) subsequent to disruption of the ADP facility and continuing until the facility is sufficiently restored.

Contingency Plans

Plans for emergency response, backup operations and post-disaster recovery maintained by an ADP facility as a part of its security program.

Emergency Response

A response to emergencies such as fire, flood, civil commotion, natural disasters, bomb threats, etc., in order to protect lives, limit the damage to property and minimize the impact on ADP operations.

Recovery

The restoration of the ADP facility or other related assets following physical destruction or major damage.

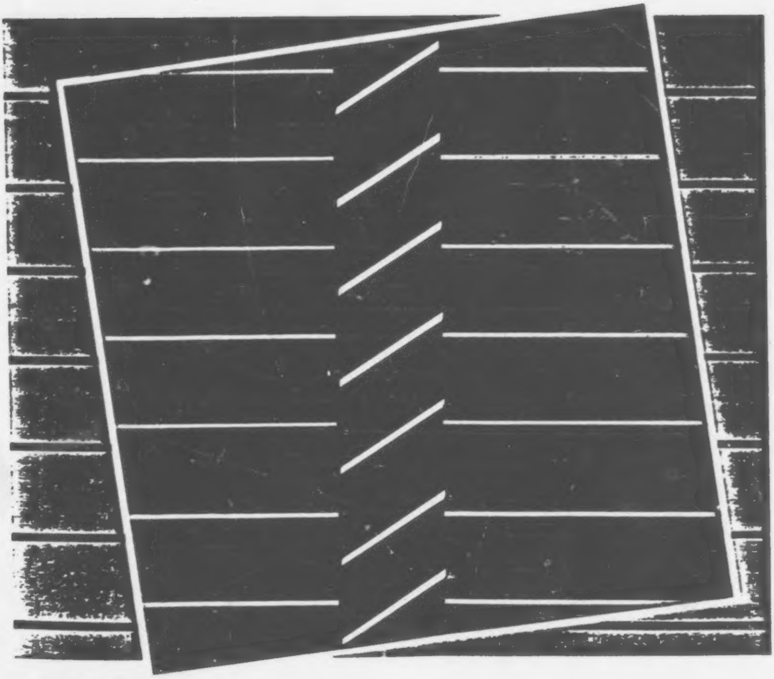
Risk Analysis

An analysis of system assets and vulnerabilities to establish an expected loss from certain events based on estimated probabilities of the occurrence of those events.

Performance Testing - Pg 116 & 118
Acceptance Testing - Pg 123

William Hetzel

The Complete Guide to SOFTWARE TESTING



QED® Information Sciences, Inc.

Chapter 7

Testing Systems—Testing in the Large

In earlier chapters I emphasized the importance of testing each phase of project work as it is carried out. With effective, ongoing testing a project will not *start* testing in the large unless reasonable confidence in the design and in the testing of individual programs has first been obtained. Without such confidence, the problems of testing in the large easily become unmanageable. Trying to test a complex system that has a shaky design and many seriously defective pieces *will not work*! This lesson has been learned over and over again.

One cannot *skip* design and program testing. All that happens when we try is that we learn the system doesn't work. Then we are driven back into cleaning up and correcting the design and the individual programs before we can continue anyway. Many programmers have tried. They are the clients I see. Bogged down in systems testing with the project 95 percent completed, they struggle to make up for inadequate early testing. The best they can hope for is excessively high system testing cost. The worst is project failure.

Despite the lessons we keep relearning, most projects don't begin serious formal testing until coding is nearly completed. The reasons are hard to understand, but certainly procrastination and misplaced optimism play important roles. We want to believe that we're doing such a good job that early testing is unnecessary. Since we don't "have" to test in the beginning, we put it off and wait until it is forced on us. Analyst availability plays a part as well. In

the early work on a project the analyst's primary concern is design work and preparing program specifications so programming can start. Only after that does the analyst begin to think seriously about testing, and by then it is too late, coding is already under way, and by the time the test plan is done, all it is able to address is testing in the large.

Meaning of Testing in the Large

In this chapter we will examine techniques and guidelines for effective testing in the large, which means any testing of *groups* of programs together. Various projects use different names for parts of the overall testing in the large effort. Examples include component testing, functional testing, systems testing, product testing, delivery testing, stress testing, parallel testing, field testing, acceptance testing, and so forth. Most organizations and designers of major projects do prepare some form of plan for this testing in the large effort. The plan establishes the testing approach and usually defines at least three distinct levels of testing to be performed. The most common names for these levels of testing are *integration testing*, *systems testing*, and *acceptance testing*.

Integration testing centers on getting programs assembled into components or functional pieces, systems testing on the functional performance of the total system, and acceptance testing on the readiness for system installation and production use.

Major Testing in the Large Activities

Name	*Purpose*
Integration testing	• Interfacing of programs and modules • Assembling system components • Program/module interactions
Systems testing	• Functional performance • System capabilities
Acceptance testing	• System requirements • Implementation readiness

In some projects, a clear distinction between these levels may not exist. For example, in small projects it is common to find only systems testing and acceptance testing (integration is treated as a part of the systems test). In larger projects the three levels may be repeated for different parts of the system (components or builds) and additional testing steps, involving quality assurance and independent testing and validation efforts, may be defined. Here I treat all of this testing under the one broad heading of testing in the large and the planning effort for all of it as the preparation of a key document I call the *Master Test Plan.*

Objectives for Testing in the Large

The first step in any good plan is to establish clear objectives. For testing in the large our key objective is to determine the overall system performance. Two basic questions need to be answered:

What is the system quality?

- What is the system capable of doing?
- What are the limitations?

Will the system support operational use?

- Are major failures unlikely?
- Is quality adequate for use?
- Is there sufficient readiness to use the system?

Answering these questions is essential to support the higher objective of making a good decision about installing a system. All of our testing is but a means to that end. While that should be obvious, I find it is easily forgotten in the heat of system testing. The goal is not to "complete" the testing, but to obtain enough confidence about the product to be able to use it safely!

Specific objectives for testing in the large have to be carefully established and agreed to by all key parties. What level of quality is good enough? How much of the system has to be working before implementation can occur? Which requirements and test cases have to be working? Which parts are less critical and may be deferred? What amount of implementation risk is acceptable?

Guidelines for Achieving Good Testing Objectives

- Start early—the first draft of the acceptance test should be prepared during requirements definition.
- Write down some tentative objectives and let people *see* them.
- Prepare a Master Test Plan and have it reviewed by project designers and end users.
- Be receptive to critiques and willing to modify objectives as the project proceeds.
- Document all agreed-upon objectives in the Master Test Plan.

These questions are a major part of the already very sizable testing in the large planning effort. They are not easy questions to resolve. Most project managers avoid answering them and are forced into making the implementation decision "in extremis." The purpose of the Master Test Plan is to help raise such issues early enough that they may be dealt with effectively.

The key is to document the objectives and make them visible for review and discussion. This is the fundamental role of the Master Test Plan and because of this document's importance to testing in the large I discuss it in much more detail in the next section.

Master Test Plan: The Central Tool for Managing Testing in the Large

I have emphasized the importance of good planning to effective testing. While unit test plans were recommended as good practice for testing in the small, the careful preparation of the Master Test Plan is an *essential* prerequisite to preparing for and managing the testing in the large effort. Unplanned and ad hoc systems testing leads to almost certain disaster. Careful advance planning is no guarantee that all will proceed smoothly, but is a *necessary* step that greatly improves the odds.

The Master Test Plan, or MTP, is a formal document that defines the testing objectives, establishes a coordinating strategy for all testing, and provides the framework for detailed planning of all steps and tasks in the process.

The Master Test Plan—What Is It?

A document (or series of documents) that is outlined during project initiation and is expanded and reviewed during a project to guide and control all testing efforts.

The Master Test Plan—Why Have It?

The primary means by which testing may be directed and controlled

- Raises testing issues
- Defines testing work
- Coordinates testing efforts
- Assigns and obtains resources

The MTP is the primary vehicle for shaping testing strategy as well as documenting the agreed-upon plans. Its contents will vary significantly, depending on the organization and the project. Some companies have broad, established testing policies that apply to all projects. In such cases, the MTP simply refers to the procedures and extends them as necessary. In others, the MTP must be more extensive. Project variables will also affect the contents significantly. A typical table of contents for an MTP has been included.

The Master Test Plan—Typical Table of Contents

Testing Strategy/Approach

- Testing life cycle definition
- Overall testing approach
- Major tools/techniques to be employed
- Definition of components and test pieces or builds
- Procedure to revise/expand and review

Test Case Specifications

- Overall organization of test cases
- Test case folios
- Requirements validation matrix
- Functional test case matrix
- Other test data information

Responsibilities

- Testing organization
- Resources and schedules
- Tasks and assignments

Procedures

- Test procedures
- Problem reporting and tracking
- Retesting

Controls

- Changes
- Reports and information

A major contribution provided by the Master Test Plan is the definition of the *Testing Life Cycle.* The concept is simply that at each step of development there are key testing deliverables; the MTP must define those deliverables and establish clearly who is responsible for each one. Chapter 3 contains a discussion of the testing life cycle and an outline of a typical set of testing deliverables. Each of the deliverables represents a major task or milestone within the overall testing effort. Taken collectively the testing deliverables

---

Putting the Master Test Plan Together

A Strategy for Inducing Commitment

1. Start with a *tentative* testing strategy for discussion.
2. Outline several alternative strategies for comparison.
3. Open Presentation/Discussion
   - Present the alternatives
   - Show testing life cycle implications
   - Stimulate *open* dialogue and comment
4. Analysis of Expected Demands
   - What is required?
   - How is each area affected by the tentative strategy?
   - What are the likely impact and implications?
   - Statement of needs, problems, suggestions
5. *Revise* the tentative strategy on the basis of discussion and analysis.
6. Statement of Commitment
   - From each unit
   - What commitments will they make?
   - How can they help?
   - What help will they require from others?
7. Identify major testing deliverables.
   - What testing work has to be performed?
   - Initial test objectives
8. Establish major test responsibilities
   - Use the statements of commitment
9. Publish initial Master Test Plan for review.
10. Review formally—test the test plan.

---

define the testing work, and documenting it in the MTP provides the vehicle for managing its completion.

There is no current standard for which testing deliverables should apply to any particular project. Over the last decade the development life cycle has matured and there has emerged a broad consensus among project managers as to what the major deliverables are. The testing life cycle is still in its infancy, and much debate can be expected over the "best" way to define and break up the work. Initiating this debate is one of the main purposes of the MTP document—the starting point in the overall testing strategy. Just as I recommended the analysis of *alternatives* when reviewing designs, I believe several testing strategy alternatives should be outlined and the impact on the testing life cycle displayed. Key individuals should then review these alternatives and help establish the best approach. Once a consensus has been reached, the first draft of the MTP can be issued and used to record the group viewpoint or stimulate further discussion as needed. The emphasis here is on *inducing commitment* from important project players and end users. The testing strategy must be understood and supported. To do that people must get involved in shaping the strategy and come away with some sense of ownership or feeling of personal contribution.

Once the initial draft of the MTP has been published and reviewed, there is an established framework for more detailed test planning. All planning information is available in one place and project participants know where to find it and what to expect. Supplementary information is issued throughout the project as more detailed plans are developed.

These supplements may come out as separate documents that refer back to the coordinating MTP, or they may be provided as updates and extensions. Typically, separate documents will be provided to detail the integration test, systems test, and acceptance test steps of the testing in the large strategy. Separate documents for individual test cases (or groups of cases) called test folios are also commonly employed, and techniques for constructing these are discussed next.

Test Case Specifications and Folios

Specialized techniques for organizing test data for program and unit testing are not necessary. The number of cases used to test a program is usually modest—five to twenty is typical—and the amount of data in each is not extensive. As a result, test cases are usually documented as a part of the unit plan or as a natural extension of the program specifications.

When testing in the large the situation is very different. The problems associated with specifying tests and keeping track of all information related to a given case are burdensome. Careful planning and organization are needed to avoid seriously impacting the quality and/or cost of the testing work. It is not uncommon to have hundreds, or even thousands, of cases and situations. For each we must have detailed procedures and specifications that define the test and tell how it is run as well as the current status. (For example: Have test data been developed? Has the test been run? When was the last time it was run? What were the results?) Tests at the systems level involve many more data, and fully specifying even a single test may require pages of output. Multiplying by hundreds of tests creates a very large data management task that is a critical part of the testing effort. *Test case folios* and *test information systems* are important tools in addressing this need.

The test case folio is simply a folder containing information on a case or related group of cases. It may be created to bring together the tests relating to a specific function or system requirement or any other convenient grouping. Once the folio is organized, *everything* related to the test cases is stored in it.

Test Case Folios—Possible Contents

1. Identification for the test group or test case
2. Testing objectives (what the test accomplishes)
3. Description of the situation being tested
4. Test data (or a reference to where it is or how generated)
5. Expected results (what the anticipated result is)
6. Entrance criteria (what has to happen before the test can start)
7. Setup and execution procedures (detail of how test is conducted)
8. Exit criteria (what it takes to sign off the test)
9. Responsibility (who is responsible for the test)
10. Activity log (record of executions or changes)
11. Actual results

All of this information is important. Because items 6 to 11 may appear to be an overkill to some practitioners, they deserve further comment. Entrance and exit criteria spell out what has to happen before a test can start and what has to be confirmed before it can be successfully signed off. For complex test situations, such pre- and post-conditions are far from obvious. Entrance criteria can be used to specify how much of the system must be present (i.e., which modules and functions) or what environment is required (daytime, test system, version, change levels, etc.). Exit criteria define the checking or verifications that must be made after test execution. They may specify certain data base or file conditions, balancing requirements, and so forth.

Setup and execution procedures define *how* the test is to be conducted.

This section of the folio can specify session sequences or refer to utility programs and procedures that must be run to conduct the test. The responsibility block defines who is responsible for the case and who is authorized to change any parts of it. Changes occur in test folios much as they do in programs. Data may have to be augmented or modified, test procedures may be changed, entrance and exit criteria altered; even the test objectives and expected results will occasionally undergo change. This has led some organizations to adopt change control procedures or version control on the folios. Thus we have version 5 of test case 38, and so on. The activity log is used to record all changes made into the folio and keep track of every attempted execution. Actual results may be stored in the folio as output listings and recorded in the activity log to maintain a complete audit trail of the results for the case. Many practitioners maintain only the most recent test run output. Practical experience has shown a need to maintain all the listings. One common example occurs when a test is suddenly discovered not to be working and we want to know how far back the problem can be traced. Other similar needs arise during retesting and debugging.

The benefits of using test case folios are substantial and grow with project size and complexity. A case can be made for retaining all of the folio contents and certainly for the use of the folio as a basic organizing tool. Besides helping to organize the testing information, the folios provide a consistent documentation format and help encourage careful planning and design of each case. Some companies have automated the folios and built what might be called a *test data base* or *testing information system*. Such systems provide a facility for entering and storing all the folio information in on-line files. Reports and displays permit easy access to the information as needed during the project. The resources used in testing are also captured and included in the data base so the system can provide broad management reports that show testing status and costs and maintain an ongoing project record.

Unfortunately, such systems are not readily available in the software marketplace. Any such system must interface with existing project control and development tools, a requirement that tends to force heavy customization. Companies that have automated test systems have built their own. One common approach is to start with a data dictionary or a library support system and extend it to support test case information. Software products for this application should become more readily available within the next few years.

Integration Testing

Testing in the large begins when we first combine programs. A group of programs must interact reasonably well before it makes sense (or even becomes possible) to test the group functionally. The objective of integration testing is to test program interfaces and confirm that programs have been linked together correctly. Our concern is not with function or requirements during this early testing in the large; instead, our goal is to ensure that we have a basic *skeleton system* that will serve as our test platform. Fatal errors such as improper calling sequences or mismatched parameters and data must be found and corrected before any cases can produce meaningful output.

Integration Testing Objectives

- Obtain a working skeleton as rapidly as possible
- Establish confidence that the skeleton parts interface and link correctly
- Demonstrate that simple test cases and transactions are being handled properly by the system

---

Most practitioners have little appreciation for the considerations involved in planning effective integration testing. Some of the considerations that integration test planning must address are listed.

Considerations for Integration Test Planning

What modules should be assembled first?

How many modules should be assembled before integration testing starts?

What order should be used to integrate the modules?
- Top levels working down
- Critical modules first
- Bottom levels working up
- Functional groups

Should there be more than one skeleton?
- How is each skeleton defined?
- Are there distinct build levels

How much testing should be done on each skeleton?
- What tests should be conducted?

These questions have no simple answers. Answering them requires a weighing of trade-offs. Any time we break down the testing task into pieces and levels we add some additional testing work. We do this in the belief that it will save work in the long run. We have learned through practical experience that attempting to test too much at once complicates the testing job and may greatly enlarge debugging and error correction time. The optimum selection of modules to be integrated and tested together cannot be made by consulting any formula—it is inherently dependent on the individual project and must be examined by looking at the pros and cons of different alternatives. Such analysis is what good integration test planning rests upon.

To illustrate, consider just the choice of the *order* of integration. We assume we have already made the decision that a certain group of modules should be tested together as a single skeleton or component.

The question is, Which modules should we start with first? All of the six listed alternatives are choices made in practice.

Some Alternatives for the Order of Module Integration

1. *Top-level modules*—Start with the driver and command or top-level modules and then work down by "plugging in" the additional modules.
2. *Critical modules*—Start with the critical system modules and integrate them, then add the rest of the skeleton around them.
3. *Bottom-level modules*—Start with the individual programs as they complete unit testing and integrate by working up to build bigger and bigger pieces.
4. *Functional modules*—Select a specific function and integrate the modules needed for that function, then proceed to the next function, and so on.
5. *As-available modules*—Take the modules that are ready and fit them together as much as possible.
6. *Complete skeleton*—Integrate all the modules in the skeleton at once and hold off any integration testing until all are interfaced.

Many combinations of these six choices are also possible integration order alternatives. For example, number 1 may be combined with numbers 2, 3, 4, or 5. Once the top level has been integrated, critical components, bottom-up components, functional components, or as-available modules can be put together and then plugged into the top level structure as a unit.

Choosing one of these orders as "best" for a particular project involves looking at the cumulative impact of many factors. The amount of "scaffolding" code that has to be constructed is one such factor. By scaffolding code we mean extra code that has to be written to support the testing process. This includes test driver or stub routines and many special programs that are written to set up test conditions or check testing output. Just as scaffolding on a building is temporary and not part of the final construction, scaffolding code is eventually removed and not a part of the final product.

How much scaffolding code is needed depends greatly on the integration alternative selected. For example, bottom-up integration requires test driver code to exercise low-level routines. Top-down integration requires stub routines to simulate missing modules. The total amount of scaffolding required is significant. In one large project where this was carefully measured, it exceeded 20 percent of the total delivered code! Nothing is quite so frustrating as discovering that a sticky problem is due to faulty scaffolding. Such errors are common because of the ad hoc manner in which most such code is implemented and tested. Thus scaffolding does impact project success, and integration planning should directly evaluate how different strategies of integration will affect it.

Another factor relates to project schedule. If certain modules are behind schedule and starting systems testing is a top concern, the integration test team may be forced to work with whatever modules are available. It may be critical to get something up and running, and the choice of that something will determine the integration priorities. Integrating critical or difficult components first may allow system testing to proceed independently on the most difficult parts of the system and thus speed up implementation.

Still another factor is system reliability. Top-down testing exercises high-level modules much more heavily than the low-level ones. Bottom-up testing is just the opposite. If system reliability depends on low-level modules, then bottom-up integration may be preferred. This is another argument for the critical-software-first approach. Such critical modules are usually the ones that influence system reliability most. Integrating them first reduces the risk that they will not be completed on schedule and increases their reliability.

All of these considerations and factors influence our choice of the best integration plan. My personal preference is biased toward small skeletons that are functionally oriented. I look for the *smallest* number of modules that may be combined and still tested effectively. That requires some means of input and output and at least some minimal processing capability or function. This grouping then becomes a skeleton to be integrated and tested as a unit. The methodology might be described in five steps.

Two major problems plague integration testing. The first is integrating modules that have not been unit tested. Just as a single bad apple will quickly rot the entire barrel, it takes only several poorly tested modules to ruin a solid integration testing plan. Some form of quality check must be made by the integration test team to ensure that modules are ready to be integrated. The second problem is failing to treat integration testing formally enough. Like all testing, the integration tests must be designed and planned to ensure

Suggested Integration Testing Methodology

1. Select a minimal group of modules that support some basic function and are not too difficult to test as a unit.
2. Link the module group together to form a skeleton and get the basic function running.
3. Test the integration of the skeleton.
    - Exercise all input and output parameters of each module.
    - Exercise all modules and calls.
    - Exercise all program options and special utility routines.
4. Stress test the skeleton.
    - Test load
    - Test performance
    - Try to break the skeleton
5. Then integrate the next group of modules.
    - Repeat steps 1 to 4 as necessary to complete the build level

thorough coverage. All interfaces should be checked, not just a random few. Systematic completion criteria such as those shown in the suggested methodology (exercising all input and output parameters, exercising all calls, etc.) need to be established and used to define the integration testing work and determine when it is completed.

Organizations that experience trouble with integration testing can expect to trace their difficulties to one of these two problems. The rush to get systems testing started makes us fall prey to accepting poor unit testing and to integrate with little planning or formal interface testing. Making the effort to test the integration carefully guarantees that we have a properly assembled skeleton and provides the platform needed to conduct our systems testing. It is worth it in terms of the trade-off of cost versus quality in our final product!

Systems Testing

The systems test starts when integration testing has been completed. It ends when we have successfully determined what the system capabilities are and corrected enough of the known problems to have confidence that the system is ready for acceptance testing. Systems testing is the major piece of testing in the large. It consumes the bulk of the testing resources and careful planning, and control of its conduct is essential to a successful outcome.

When System Testing Begins and Ends

System testing *begins* when a minimal system or skeleton has been integrated

System testing *ends* when we have measured system capabilities and corrected enough of the problems to have confidence that we are ready to run the acceptance test

---

The Master Test Plan and test case folios have already been discussed as important tools in planning and organizing the systems test. In a real sense everything I have talked about in the book has been motivated by lessons learned from systems tests that have gone awry. The systems test for a large, complex project is a large and complex task. The emphasis on testing and reviewing each phase of work as it is conducted, on thorough and systematic unit and integration testing, and on the Master Test Plan concept are all aimed at the prime goal of reaching the systems test in such a position that it can be completed relatively easily.

This is the payoff for all the effort we invested earlier! We want to avoid the agony so often experienced in the final throes of project implementation. We want to minimize the painful and labor-intensive process of searching for and correcting major system deficiencies. We want to avoid last-minute changes that go poorly tested. This is our motivation for improved testing processes and what makes an investment in better testing methodology produce such a high return.

The planning for systems testing starts with the functional test cases—those sufficient to demonstrate or cover the functional performance of the system. It may be well for the reader to return to Chapter 6 and review the earlier discussion of functional and black-box testing of programs. All the same principles and guidelines apply. The only difference is that we are now planning the testing for a group of programs instead of a single one. Some of the functional cases derived for program testing may be reused during systems testing. Others derive from the system requirements specifications.

As each group of cases is designed a test folio should be created for it. The folio format should specify the information needed to define each case. At a minimum this must include the test objective, test data, expected results, and any test procedures or special requirements. As mentioned earlier, entrance and exit criteria and an activity log are also very helpful.

The second major part of the systems test to be planned is the tests of *performance capability*. The functional cases concentrate on demonstrating system functional capability. Additional cases are required to measure system

---

Functional Capability Systems Testing

Purpose: Provide a *systematic* demonstration that all functions are available as specified.

Is user oriented

- No knowledge of internals
- Best done from user manuals

Key technique is careful planning

- All classes of valid input must be accepted
- All classes of invalid input must be rejected
- All functions must be exercised

Organize test folios

- Around requirements
- Around key functions
- Around special test cases
- All test cases to be documented
- All test cases to have predicted results

Look for systematic coverage

- Functional coverage matrix
- Requirements validation matrix
- Transaction flows
- Functional boundaries

Review

- To identify additional tests
- Evaluate effectiveness of current ones

[redacted]

Performance Capability Systems Testing

Purpose: Provide a systematic determination of what the performance capabilities are and demonstrate that performance and overall quality will meet the operational requirements of the system.

Is volume and limit oriented

- Drive the system to its limits
- Determine where it breaks down
- First test to specification, then break and analyze

Key technique is careful planning

- Sources of test data
- Use of simulators and test data generators
- Experiments and models

Organize test folios

- Around each performance feature or attribute
- Around each quality factor

Review

- To identify additional tests
- Evaluate effectiveness of current ones

One of the performance capability tests has been given the special name of *volume testing*. A volume test is planned with the specific objective of determining how many transactions or how many records can be operationally supported. Functional cases that have already been tested are reintroduced to the system at higher and higher volumes until the system is no longer capable of effective operation. If the test is carefully designed, it determines a range of suitable volumes for the system and measures the "breakpoint." Extracted test data (taken from live files) or generated data (either from a test data generator or replication of already defined cases) avoid the problem of having to hand prepare volume data.

Other special performance tests are gaining more common use. These include tests of system documentation (usability, completeness); system reliability (failure analysis, stress testing, operational testing); and system maintainability (documentation, time to make standard changes). Just as with the functional tests, folios for each case or case group have to be organized and formal documentation maintained to ensure that we have thought out the test case carefully and understand what it is attempting to measure.

When discussing testing in the small I placed a great deal of emphasis on structural test cases and developed the important techniques of structured basis testing to provide systematic coverage of the underlying logic. Until recently, these notions had little impact on testing in the large. The reasons were strictly practical limitations. Systems testing is applied to large numbers of programs at once, with many thousands of lines of logic. The test cases are usually constructed without the designer ever looking at the code. Systematically examining the logic to produce additional systems test cases requires too much time to be worth the effort. However, during the last few years new tools and techniques that are changing the picture have come along. I believe these will play increasingly important roles in systems testing in the future.

The concept is the same as in testing in the small. What we would like to do is select a set of test cases that systematically *cover* or explore the system logical structure. While it is generally impractical to spend time analyzing low-level logic to achieve this, there are two ways in which structural systems coverage may be obtained.

Structural Systems Testing

Purpose: Provide systematic test coverage of the underlying logical structure.

Two approaches

1. Model high-level system flow and apply structural testing ideas from unit testing to these models.
2. Instrument the system to *measure* the structural coverage obtained by normal systems testing and to identify areas requiring further cases.

The first approach is to apply the structural analysis at a high logic level. System or transaction flow diagrams offer a good illustration. It is common practice in on-line systems to define the various transactions with transaction flow diagrams. By transaction, I mean a set of operations that begins with an input and ends with one or more outputs. The transaction flow diagram is a *model* showing the sequence of steps performed in the processing of the transaction. The model illustrates what happens functionally; it does not show the detailed or actual control structure of the program. Nevertheless, we are able to apply structural testing techniques to it just as we did in program testing. A basis set of test cases that will test each process step and exercise all decision outcomes can be derived. We can also use the flow diagram to find achievable (structurally possible) flows that logically should not exist or to develop special cases for the most tortuous possibilities. All of these make excellent transaction cases. Similar types of structural analysis may be applied to system flow diagrams and high-level system design models or logic flows.

The second approach to obtaining structural system coverage is through *instrumentation*. At the simplest level we might establish the criterion that every module in the system is entered by at least one of our selected cases. We could "instrument" the system to record each module entered during testing, check this list against the list of module names, and report any modules missed. Cases would be selected as in any systems test planning effort. These cases are then executed by using the instrumented system to automatically record the extent of coverage. The method may easily be extended to test for more complete structural coverage. All interfaces might be instrumented so as to gauge the effectiveness of the integration test. If instrumentation is inserted at each program decision point, path coverage may be measured and any statements not executed by the set of systems tests determined. Some organizations have established criteria such as 90 percent or 95 percent of all system branches being exercised by the systems test cases. Use of such quantified coverage criteria provides insurance that the systems test has been comprehensive and has actually been performed.

Systems testing should produce a key deliverable called the *test data set* which is a special set of cases designed to exercise a system over most or all of its capabilities. Its purpose is to support retesting when system changes are made. The cases are normally a subset of those used in the systems test. The logical time to produce the test data set deliverable is during systems testing. All that is required is to "save" the cases we have constructed so that they will be available for reuse as changes to the system occur. Preparing the test data set as a by-product of systems testing costs little; preparing it at any other time is usually so expensive that we end up not having one at all!

The System Test Data Set

- A special set of test cases designed to exercise a system over most of its capabilities
- Prepared as a by-product of the systems test
- Used to retest the system after any changes are made to it
- Output compared to show performance not affected

I shall have much more to say about the test data set in the next chapter. The key consideration here is to plan to produce one as a part of each systems test. If a test data set is produced for every system as it is developed, then a complete data base of such test files will be available for all operational systems. Such a test file is a prerequisite to high-quality, cost-effective maintenance.

Acceptance Testing

The third major component of testing in the large is the acceptance test. The purpose of acceptance testing is to confirm that a system is ready for operational use. The systems test measures and determines what the systems capabilities are. It ends when the *tester* is confident that the system is ready to be used. The acceptance test is performed for or by *someone else* (normally the eventual user) to demonstrate that the confidence is justified.

System Acceptance Test

Purpose: To demonstrate that a system is ready for operational use

Approach

- Focus is on user requirements
- A final stage of confidence building
- User involvement is major
- Is *expected* to be successful

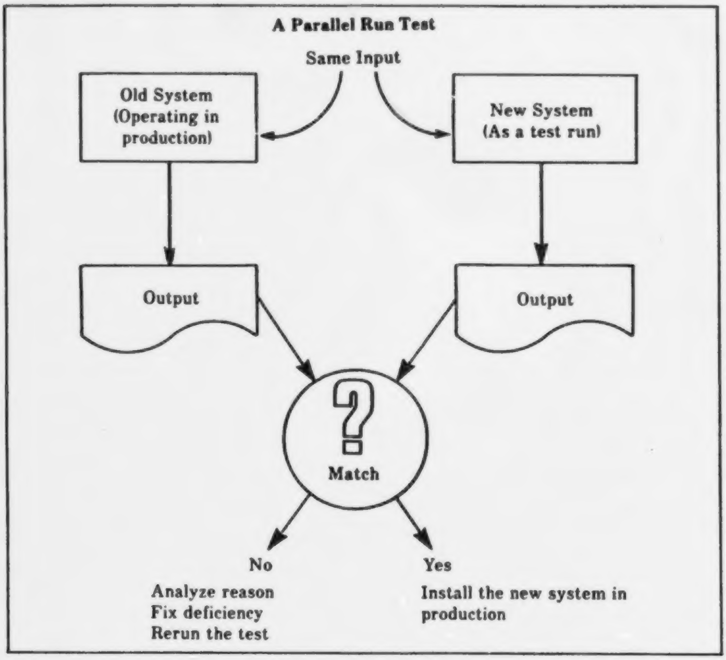
I favor an acceptance test that is expected to work and is executed smoothly and rapidly (preferably within several days). If the systems test has been thorough, there should be few surprises and virtually no new discrepancies discovered during the acceptance test. This test exists as a final stage of confidence building for the user and as protection against overly risky implementation for the organization.

The selection of the tests to include in the acceptance test should be made by the user, but with expert professional help from the test planners. Many approaches are possible. The focus is on major functional and performance requirements. If good requirements testing was done during the requirements definition phase tests will already be defined and associated with each requirement. All the acceptance test planners need do is dust them off and review or augment them as necessary. Most of the acceptance cases can be taken directly from the systems test. The only difference is that the cases will now be run in a quasi-operational environment, perhaps using the actual hardware, software, and user environment that is planned for production. Other common test approaches are to take a typical day's worth of transactions, or a hypothetical month or year of operations, and to demonstrate that all business for the typical day or hypothetical period is properly processed.

Major Sources of Acceptance Test Cases

- Requirements tests
- Subset of the systems test cases
- Typical processing day or period
- Pilot operation

When testing a replacement system, the *parallel run* is a commonly used strategy. The "old" system processes a selected amount of data (day or month, for example) during its routine production run. These data are saved and reentered as input to the "new" system to be processed again "in parallel." The output is compared and balanced against the parallel run output to confirm proper processing.


A Parallel Run Test
Same Input
Old System (Operating in production)
New System (As a test run)
Output
Output
?
Match
No
Analyze reason
Fix deficiency
Rerun the test
Yes
Install the new system in production

[redacted]

Example of Acceptance Test to Test Endurance

- Test will be run on operational hardware and software.
- Test will contain periods when system is stressed significantly.
- Duration will span the full run cycle. (For continuous systems this should be at least twenty-five hours.)
- All interfaced systems must be in operation throughout the test.
- [redacted]
- All major functions and interfaces will be exercised during the endurance run.
- Not completing the entire test for any reason requires the full test to be restarted.

Similar formal tests should be defined for each major requirement or performance criterion. Test folios for each case selected are as important to acceptance testing as for any other testing in the large activity. Special emphasis needs to be given to defining test objectives, and "success" criteria and all the important quality attributes need to be acceptance tested, including reliability, maintainability, and documentation.

Organizations experience several common problems with acceptance testing. The biggest is forgetting that a *perfect* system is not the goal! We all know that every system has deficiencies. We don't want to wait until all of these are corrected before we begin using the system. Rather, the system should be accepted if it is good enough to use. The primary issue is usability and reliability. Will the system support operational use? Thus the acceptance test should be designed to focus on usability and reliability and not get bogged down with a mass of minor discrepancies and problems. It is fine to keep track of such minor problems, but they must not interfere with the broader objective of the test or adversely affect the system's acceptance.

A second major problem occurs when the acceptance test responsibility is placed totally in the user's hands and the user is not equipped to carry it out.

Two Common Pitfalls in Acceptance Testing

- Forgetting the objective—striving for perfection instead of effective operational use
- Testing without planning—informal playing with the system by the end user as a substitute for true acceptance testing

Planning effective testing takes an experienced and knowledgeable professional. Novice users cannot be expected to know the importance of a careful test plan or of carefully documenting expected results for each test case. Their testing takes the form of "informal system exercise." They try some cases to see what happens. If things "look" all right they accept the system; if they don't, they reject it. This is a poor approach to acceptance testing and may be likened to the situation of a teacher deciding on what it takes to pass an exam after the test is over.

Automated Tools and Aids

Software tools and support packages are very important to testing in the large. Knowing what tools are available and how to select and use them impacts the testing in the large plan and can greatly influence effectiveness and cost. The most important tools include instrumenters, comparators, test data generators, simulators, debugging and documentation aids, and test case management or testing information systems.

Important Tools for Testing in the Large

- *Instrumenters*—to measure test coverage
- *Comparators*—to compare outputs, files, and source programs
- *Test data generators*—to generate files and test data input
- *Simulators and test beds*—to simulate complex and real-time environments
- *Debugging aids*—to aid in determining what is wrong and getting it repaired
- *Testing information systems*—to maintain test cases and report on testing activities
- *System charters and documenters*—to provide automatic system documentation

I gave examples of many of these tools in the last chapter. (The reader should refer back to Chapter 6 for examples of how instrumenters, comparators, and test data generator packages operate.) While helpful in unit testing situations, the tools are often indispensable for large-scale testing.

Many situations arise when it is necessary to build large test or conversion files quickly. Test data generators serve that need ideally. They also facilitate volume and performance capability testing. Planning for testing in the large should always consider their potential use carefully. Properly employed, they may save enormous amounts of time that is otherwise spent laboriously constructing test data.

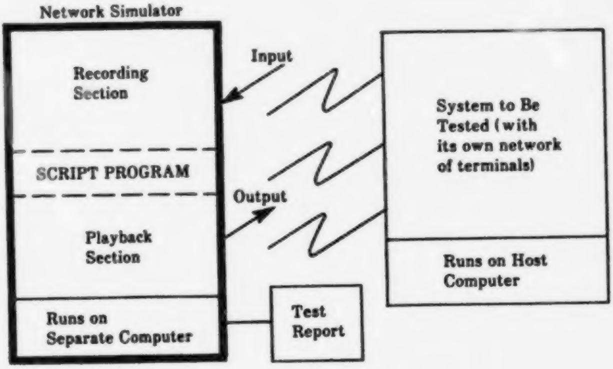
The instrumenters are equally important. Test coverage may be measured in the large, just as it is in the small. In many ways the measurement of coverage in the large is more informative because other information about the extent of structural testing is not available. The SOFTOOL package described in Chapter 6 has good facilities to support such measurement. The program or system level report shows the programs executed during the last test and over all tests in a given test series. This allows confirmation that the system test set invokes all programs and provides documentation that the tests are actually executed. The same capability is available at the paragraph level and the statement level. If considered as the system is constructed, a self-instrumenting capability is easy to build in. As a simple example, each program or paragraph may be built with a compile time option to write a record to a special file each time it is invoked. Processing this file will give an effective measurement of test coverage without requiring the purchase of any specialized packages. Other similar self-measuring techniques are in common use.

As with test data generators, proper use of comparators can save a great deal of time and improve system testing effectiveness. Planning for their use is the same as that discussed in the last chapter. The decision to use the comparator should be made when the test case is designed. Testing procedures and the definition of expected results can then be constructed with the comparator tool in mind.

Simulators and test beds are two additional tools of significance to testing in the large. Two examples suffice to demonstrate how knowledge and use of the right simulator may totally alter the testing in the large plan. Consider first the problem of testing a large on-line system with many thousands of terminals dispersed geographically in a large network. There is no practical way to control input at the dispersed terminals so that, without a simulator, testing must be conducted from a few selected test terminals. Such testing cannot effectively measure response times, yet for volume or stress testing that is the key issue! The solution to the problem is the use of a network simulator.

The diagram shows how the typical network simulator operates. The simulator can be connected to an existing network and instructed to record actual network transactions as they occur in real time. In effect this is a legal "wiretap" to provide a permanent record of everything that occurred at every terminal in the system. Once this is recorded, the system may be tested by

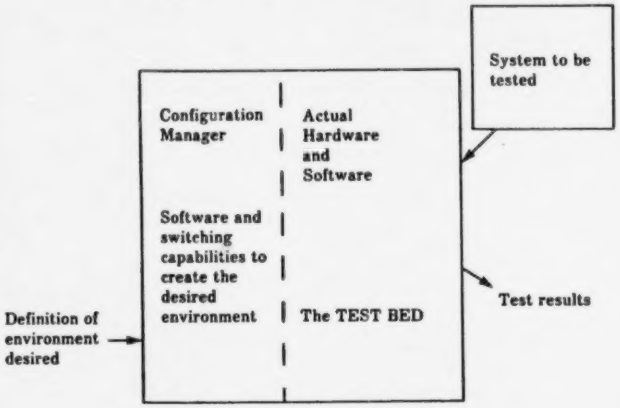
Example of a Typical Network Simulator


Network Simulator
Recording Section
SCRIPT PROGRAM
Playback Section
Runs on Separate Computer
Input
Output
System to Be Tested (with its own network of terminals)
Runs on Host Computer
Test Report

instructing the simulator to play back the transactions just as they occurred. To the system this looks just like a typical business day, with thousands of on-line terminals sending in their transactions. Even line drops and line failures are replicated just as they occurred in real time. Since the simulator is running on its own computer, the response times in the test system are realistic. By supplying a "script" of special instructions to the simulator we are able to play back the transactions twice or ten times as fast, or select special transactions, define new transactions, and so forth. This allows us to design comprehensive tests and exercise the system being tested at steadily increasing levels until we learn where it breaks or where response time reaches unacceptable levels. Such superb testing capabilities are simply not possible without the use of this type of support tool.

As a second example, consider the problem of testing a system that requires operation in dozens of different hardware and software configurations. This is a case where an advanced test bed system or environment simulator (see figure) is ideal.

Example of an Environment Simulator (Test Bed)


System to be tested
Configuration Manager
Actual Hardware and Software
Software and switching capabilities to create the desired environment
The TEST BED
Test results
Definition of environment desired

Input to the environment simulator is a description or definition of the particular "test bed" or hardware/software environment desired. The configuration manager "creates" this environment by selecting the appropriate system and making it available for the system to be tested. A test can then be run by using one environment; then the environment can be "changed" and the test run again. Comparators compare the test results and report any discrepancies.

These examples are two of the more common uses of simulators in large-scale testing. Many other powerful simulators are available. These are often industry specific, and testing planners must take special efforts to learn what is available and how it may be used to simplify or improve the effectiveness of their testing.

Debugging aids and system documenters are also important to the testing in the large effort. The many diverse system charting and automated documentation tools help the tester to design tests and understand the overall system. The debugging aids are designed to help discover the underlying problems or causes of discrepancies discovered during testing. Although both the documenters and the debugging aids are, strictly speaking, not testing tools, testing practitioners must be familiar with them and often will have to use them in carrying out their responsibilities.

A final tool of major significance is the testing information system or test data manager. These were discussed earlier in this chapter in the section on test case folios. They offer facilities for maintaining and tracking all data and provide testing status and performance reporting. In general, such systems are not available as off-the-shelf packages (at least at this time). The small number of companies that have such capabilities have constructed them for themselves.

Tracking and maintaining familiarity with all the tools mentioned in the last two chapters is not easy. The tools are proliferating rapidly, and many of the best ones are not promoted in the public marketplace. If you are considering tools in your organization, this may be an area in which consulting help is advisable. Many catalogs and directories are available. Reifer, Inc., and SRA are two companies that sell a specialized catalog of software tools, and many of these are of direct interest to testing practitioners. The National Bureau of Standards is involved in a major project to build and maintain a tool data base. The Datapro and Auerbach manuals are other excellent sources (at least for software that has been successfully marketed to a number of active users). Finally, a number of companies offer a data base search for software products. SofSearch manages a data base of over forty thousand different software products that permits users to search and receive hard-copy reports within forty-eight hours. All of these sources, together with user groups and professional associations, provide excellent means of staying abreast of what is available.

The benefits of selecting the right tools are major. *Good* tools produce significant productivity gains and can strengthen testing effectiveness sharply. They often help in building the testing discipline and tend to have a multiplier effect, brought about through the synergy of people and machines working together effectively. Improper care in selecting the tools and getting them implemented is guaranteed to produce failure and result in another package that lies around gathering dust. There are many such failures. To avoid them you must pick a tool that is easy to use and integrates easily into the existing development and testing methodology within the organization and then you must provide training and help the practitioner to use the tool properly. Without support, even the best tools wither away and fall into disuse. The true total costs of acquiring a package tool are usually greatly underestimated. The cost of selecting a tool, getting it installed, operating it, and supporting it properly over its lifetime is likely to be at least ten times the cost of the license fee. While the fee gets all the attention, it is only a small part of the total cost of acquiring a new tool.

SUMMARY

The cornerstone for effective control of testing in the large is the Master Test Plan.

The Master Test Plan is the integrating vehicle for all project testing issues. It provides

- Strategy and approach
- Testing life cycle definition
- Test case specification
- Responsibilities
- Procedures
- Information and controls
- Tools and aids

The Master Test Plan is an essential document

- Crucial to prepare carefully
- Crucial to review
- Crucial to maintain

It must be drafted early in the development process (preferably during requirements definition or early design)

Testing in the large cannot cover up ineffective design and unit testing.

The most effective way to improve testing in the large is to start with good unit testing.

Testing in the large generally consists of at least three distinct activities.

- Integration testing
- Systems testing
- Acceptance testing

Two basic questions must be answered by the testing in the large plan:

- What is the system quality?
- Will the system support operational use?

The concept of a testing life cycle is that there are key testing *deliverables* to be produced during each stage of software development.

The purpose of the Master Test Plan is to shape and define the testing life cycle.

Test case folios provide an organized means of *documenting* and *maintaining* all case information and specifications.

Test case folios form the basis for an automated data base or testing information system.

Integration testing serves to confirm that we have linked up a basic system suitable for testing.

The approach to integration has a significant impact. It must be carefully planned and evaluated.

A suggested integration test approach is to:

- Select *small* functionally related groups of modules.
- Get a basic or simple transaction running.
- Stress-test the integrated modules to confirm a solid test platform.

Systems testing begins when a minimal skeleton is integrated.

Systems testing ends when we feel confident that the system will pass the acceptance test.

Systems testing depends on good planning

- Functional capability tests
- Performance capability tests
- Specialized tests (volume, documentation, etc.)
- Structural coverage measurement

Systems testing must produce a test data set for subsequent modification testing as a by-product.

Acceptance testing serves to *demonstrate* that a system is ready for operational use.

Acceptance testing must be planned and formal.

A vital part of the Master Test Plan is the selection and effective use of test tools for all phases of testing in the large.

Major tool categories of importance to testing in the large include

- Instrumenters
- Comparators
- Test data generators
- Simulators and test beds
- Debugging aids
- System charting and documentation aids
- Test case management systems

The total cost of a software tool is many times the license fee or purchase price.

THE WALL STREET JOURNAL MONDAY, NOVEMBER 25, 1985

A Computer Snafu Snarls the Handling Of Treasury Issues

By PHILLIP L. ZWEIG
And ALLANNA SULLIVAN
Staff Reporters of THE WALL STREET JOURNAL

NEW YORK—A computer malfunction at Bank of New York brought the Treasury bond market's deliveries and payments systems to a near-standstill for almost 28 hours Thursday and Friday.

Although bond prices weren't affected, metal traders bid up the price of platinum futures Friday in the belief that a financial crisis had struck the Treasury bond market. However, Bank of New York's problems appeared to be more electronic than financial.

The foul-up temporarily prevented the bank, the nation's largest clearer of government securities, from delivering securities to buyers and making payments to sellers—a service it performs for scores of securities dealers and other banks.

The malfunction was cleared up at 12:30 p.m. EST Friday, and an hour later the bank resumed delivery of securities.

But Thursday the bank, a unit of Bank of New York Co., had to borrow a record $20 billion from the Federal Reserve Bank of New York so it could pay for securities received. The borrowing is said to be the largest discount window borrowing ever from the Federal Reserve System. Bank of New York repaid the loan Friday, Martha Dinerstein, a senior vice president, said.

Although Bank of New York incurred an estimated $4 million interest expense on the borrowing, the bank said any impact on its net income "will not be material." For the first nine months this year, earnings totaled $96.7 million. Bank of New York stock closed Friday at $45.125, off 25 cents from Thursday, as 16,500 shares changed hands in composite trading on the New York Stock Exchange.

Bank of New York said that it had paid for the cost of carrying the securities so its customers wouldn't lose any interest.

Bank of New York's inability to accept payments temporarily left other banks with $20 billion on their hands. This diminished the need of many banks to borrow from others in the federal funds market. Banks use the market for federal funds, which are reserves that banks lend each other, for short-term funding of certain operations. The cash glut caused the federal funds rate to plummet to 5⅛% Friday from 8⅜% early Thursday.

The electronic snafu is by far the largest of computer problems that periodically have bedeviled the capital markets.

Almost all government securities transactions are settled electronically through the New York Federal Reserve Bank. In this system, computers of clearing banks are linked to one another through a central computer, to enable banks to settle purchases and sales of securities by customers.

According to Wall Street sources, the malfunction occurred at 10 a.m. Thursday as Bank of New York was preparing to change software in a computer system and begin the day's operations. Until Friday afternoon, Bank of New York received billions of dollars in securities that it couldn't deliver to buyers. The Fed settlement system, which officially closes at 2:30 p.m., remained open until 1:30 a.m. Friday in the expectation that technicians would be able to solve the problem.

Rumors about bank problems often send commodity traders scurrying to buy precious metals. In the platinum pit at the New York Mercantile Exchange, the price for January delivery surged $12.40 an ounce to $351.20 Friday on volume of 11,929 contracts, a 29-year record.

Reports that the Fed was investigating transfer problems at Bank of New York prompted the platinum buying.

American Banker

November 25, 1985

Bank of NY Computer Failure Disrupts US Securities Market

Bank of NY Said to Borrow $24 Billion Overnight from Fed

NEW YORK — Serious computer problems at the Bank of New York last Thursday disrupted the government securities market and forced the bank to borrow more than $20 billion overnight from the Federal Reserve Bank of New York, according to industry sources.

The computer failure, involving the bank's link with the government securities network operated by the Federal Reserve System, prevented the bank from transferring upwards of $20 billion in U.S. Treasury instruments, sources said. Some estimates ranged as high as $60 billion. The Fed's securities wire was kept open until 2:15 a.m. on Friday in an attempt to process as many transactions as possible, the sources said.

This story was reported by Tom Ferris, Robert M. Garsson, and Bart Fraust. It was written by Mr. Ferris.

Unable to complete all its trades, the bank said it needed "to borrow a large sum overnight from the discount window" of the New York Fed to meet its reserve account requirement. Industry sources put the loan at $24 billion.

The computer problems continued into Friday, and the New York Fed informed its member banks at 11:45 a.m. that securities transactions destined for the Bank of New York would not be accepted. That suspension was lifted at 1:31 p.m. after the Bank of New York informed the Fed it had cleared up the "system malfunction."

A Federal Reserve spokesman said it was unclear whether the wire would close on time Friday.

In a statement issued Friday afternoon, the Bank of New York said its borrowings from the Fed were "fully collateralized in accordance with Federal Reserve policies." The bank said the loan was being repaid Friday "as the securities are delivered in the ordinary course of business."

The bank also said that "the financial impact on the company's net income will not be material."

However, industry sources said the malfunction was likely to be relatively costly to the bank. Besides the expense involved in going to the discount window for such a huge loan, the bank will have to pay one day's interest on trades it was not able to complete, these sources said.

An overnight loan of $24 billion borrowed at the 7.5% discount rate would cost about $4.9 million. The Bank of New York had earnings of $107.5 million in 1984.

In New York Stock Exchange trading, Bank of New York shares closed at 45¼ Friday, down a quarter point.

Bank of New York is one of the largest clearers of book-entry government securities. The Fed's book-entry system is fully computerized, eliminating the need for the physical exchange of securities certificates. Trades are made via messages sent over a communications network operated by the Fed, which serves as the agent for the U.S. Treasury Department.

The extent of the Bank of New York's problems was unclear Friday, and officials declined to elaborate on the bank's brief statement. A spokesman for the New York Fed also would not discuss the details of the problem.

Evidently, the bank continued to receive securities Thursday and wire funds out in payment. However, it was unable to transfer ownership of those instruments on to their purchasers via the Fed's book-entry system, and thus built up a huge deficit in its reserve account with the Fed. That deficit had to be funded through the discount window borrowing.

Such significant disruptions are rare, sources said. "It's very unusual for a bank to be down for a whole day," one banker said. Other sources could recall only one or two similar occurrences.

The incident last week also affected the federal funds market. Ray Stone, chief financial economist for Merrill Lynch & Co., said late Friday that fed funds were trading in the range of 6% to 6.5%, as much as 200 basis points lower than the 8% rate that would have been expected without the disruption.

"There's no way they should be trading in that range," he said.

The massive borrowing at the Fed's discount window had the effect of increasing bank reserves, which in turn drives down other short-term rates.

Mr. Stone said the impact of the Bank of New York system failure was limited to "a one-day movement in the funds rate. The market isn't paying any attention to it."

Meanwhile, a spokesman said the New York Fed would want to satisfy itself that the operational problems have been corrected. ■

Investor's Daily ★ Monday, November 25, 1985

Bank Of N.Y. Computer Problem Disrupts Securities Marketplace

By Anne Croly, *Investor's Daily*

NEW YORK — A computer malfunction at the Bank of New York late Thursday and part of Friday disrupted wire transfers of federal funds and government securities, sending ripples of anxiety through the marketplace.

The Bank of New York, which describes itself as the largest clearer of book-entry government securities, also clears securities transactions for a number of large New York-based brokerage houses and money center banks. The bank transacts a total of $60 billion in government securities daily, which reflects both the buy and sell sides of the transaction.

The bank blamed a "serious (computer) malfunction" for the disruption, which left it short of funds at the end of the day Thursday.

The computer problem was resolved about midday Friday and normal clearing operations resumed, the bank said. Clearing is the settlement process that matches up and verifies all buyers and sellers.

Because of the malfunction, the Bank of New York borrowed a "large sum overnight" from the Federal Reserve System's discount window, the bank said in a statement. The loan was fully collateralized, in accordance with Fed regulations, and was repaid Friday as the securities it normally handles were delivered. The financial impact on the company's net income "will not be material," the bank said.

Traders' estimates of the amount of the bank's discount window borrowings ranged as high as $20 billion, but the bank refused to comment.

Apparently on Thursday the Bank of New York took in both the securities and the payment for those securities and sent out the payments. But because of the breakdown it could not deliver the securities. For a while Friday the bank suspended acceptance of securities but rescinded that order around midday, a trader said.

Another trader noted that the system became so backed up that the Federal Reserve kept its federal funds wire open until 2:30 a.m. EST Friday in order to accommodate the financing needs of the Bank of New York. The system was still so clogged when it was finally operational Friday that the bank was still clearing up Thursday's delays, one dealer said.

This failure by one of the major clearing systems left the credit market "riddled with distortions," one trader said. Brokerage houses and banks will be uncertain of what their cash positions are, which could be reflected by gyrations in the federal-funds market for the next two-week maintenance period, he said.

In addition, the excess reserves injected into the system by the borrowings from the discount window left the federal funds market awash in liquidity, which could mean more Fed activity in days to come, bond market sources said. The federal funds rate, which closed Thursday at 7%, fell as low as 5½% early Friday.

These clearing difficulties are fairly common, however.

Another trader noted that last Wednesday the failure of the clearing system at Manufacturers Hanover Corp. forced that bank to borrow more than $4 billion from the Fed in a single day. The Federal Reserve issues these borrowing figures on a weekly basis.

American Banker

November 27, 1985

A Hard Day's Night for the Bank of New York

Relief on Wednesday Became Anxiety on Thursday When Computer Failure Snarled US Debt Trades

By TOM FERRIS

NEW YORK — In the hectic business of processing government securities transactions, last Wednesday was especially chaotic. Computer problems and heavy volume combined to disrupt operations at the nation's major clearing banks. Executives breathed sighs of relief when, several hours later than usual, the books were closed on $230 billion of transactions.

Little did they know that the next day would be far worse.

On Thursday morning, the Bank of New York, the primary clearing agent for a number of major Wall Street brokers that buy and sell U.S. Treasury instruments, was unable to start its operations. The heavy volume from the day before had "swamped" a new software program and knocked out the bank's link to the book-entry government securities network operated by the Federal Reserve System.

The problems continued all day, disrupting the massive government securities market and forcing bank officers in money centers to extend their workday past midnight. Bank of New York executives, headed by J. Carter Bacot, the chairman and chief executive, were on the phone throughout the night to officials at the Federal Reserve Bank of New York.

Early Friday morning, when it became clear that the bank's computer malfunction could not be sorted out, the securities wire was closed, and E. Gerald Corrigan, the president of the New York Fed, personally approved a $24 billion overnight loan to the bank, said to be the largest borrowing ever from the Fed's discount window.

The incident displayed just how dependent on computers the financial markets have become and how vulnerable banks are to system failures, said observers, who added that there were no assurances that the same kind of snafu will not beset a major clearing bank again. The episode will cost Bank of New York $1.5 million to $2 million after taxes, according to James McDermott, a bank analyst with Keefe, Bruyette & Woods Inc. That would amount to about a nickel or a dime a share, he said.

The crisis also gave the Fed a chance to show it would act forcefully to stem marketplace disruptions, the observers said. "Once again, the New York Fed rose to the occasion and played a critical leadership role and calming role," said a source close to the Bank of New York. Fed officials, invoking standard agreements it has with clearing banks, approved the massive overnight loan using the securities of Bank of New York's customers partly as collateral. When the computer failure was corrected on Friday, the borrowing was repaid as the delayed securities transactions were finally cleared.

Problems Focus Attention

Bank of New York's problems also brought some attention to the little-publicized but huge market for the primary and secondary trading of Treasury bills, notes, and bonds. Virtually all of these transactions are handled through the Fed's book-entry securities system, which eliminates the need for physical certificates because it electronically records the ownership of Treasury instruments. On an average day, securities valued at $230 billion change hands in 30,000 transactions, according to the New York Fed.

The market is dominated by four large New York clearing banks, which account for more than half of the entire national volume, according to a Fed document. Bank of New York alone processes about $60 billion a day, market sources said. Other big players include Manufacturers Hanover Trust Co., Irving Trust Co., Marine Midland Bank, and a New York affiliate of Security Pacific National Bank, Los Angeles, each with more than $10 billion a day.

Computer difficulties at clearing banks and at the Fed itself frequently delay processing on the central bank's securities wire and its funds-transfer network, known as the Fed Wire. If necessary, the wires are kept open, and the difficulties almost always are sorted out within a few hours.

Last Wednesday was an example. An unidentified clearing institution — not the Bank of New York — suffered systems problems that knocked it out part of the day and interrupted processing at other banks.

The unusually heavy traffic on the Fed's network that day — 49,000 trades — further complicated matters. Volume was heavy because Wednesday was the settlement date for monthly "Ginnie Mae" securities issued by the Government National Mortage Association. It was also the last day of the two-week reserves maintenance period, the time during which a bank must hold on deposit at the Fed an average amount equal its daily reserve requirement. Volume tends to be heavier on the last day of the period because banks make additional transactions to meet their reserve-account positions.

The securities wire, which usually closes by 2:30 p.m., was open until 10 p.m. last Wednesday because of the problems, said Peter Bakstansky, a New York Fed spokesman. The Fed Wire, which typically cuts off at 6:30 p.m., had to be kept open until the securities wire closed. The day's transactions were settled late but in full, and the books were closed without incident, Mr. Bakstansky said.

Thursday was another matter. The Bank of New York had trouble getting its securities-processing operations up.

Computer Failure Spotlights US Securities Trade Vulnerability

On Wednesday the bank had installed a new software program and the heavy volume overtaxed it, sources said. There were errors in the settlement of Wednesday's transactions, but the bank's operations did not break down until Thursday morning, one source said.

The nature of the computer failure was unclear. Bank of New York officials declined requests for interviews, and the bank would not elaborate on a brief statement issued last Friday. But one source close to the bank said the malfunction stemmed from the heavy processing load Wednesday. The new software program "just couldn't handle the volume," this source said. "It was jumbling everything."

Bank of New York officials assured the Fed throughout the day that they would be able to correct the glitch in time to process the day's work, according to Mr. Bakstansky. Based on past experience, it was not an unlikely assumption, he said. "We had every reasonable expectation of them coming back up," he said.

Throughout the day, banks continued to send securities to the Bank of New York, with payment automatically being made by a debit to the bank's Fed account. However, the bank itself was unable to send securities on to other banks, and, as a result, it was building up a huge surplus of securitized assets that had to be passed on to their purchasers. As the number of securities on hand increased, the balance of the bank's Fed account dropped.

It is unclear whether the bank resorted to other means to process the payments, such as manual handling or reverting to the old software system. The huge loan from the discount window, however, indicates that, if taken, any such steps were unsuccessful.

"Usually you find a way to deliver," said one banker who is an expert on the payment system. Typically, he said, a bank will take out its biggest trades and make sure they are processed to reduce the costs involved with the failure to deliver securities. "It looks like [the Bank of New York] lost control of what they were doing," he said.

Preparing for Massive Loan

Another possibility would have been preventing other banks from sending securities to Bank of New York, thereby limiting debits to its Fed account. But that is a risky proposition, observers said, because it could cause chaos at other banks. "There are broad market implications to cutting someone off," said one source. Added another: "It would raise hell in the marketplace."

By around 7 p.m., the New York Fed was preparing for the possibility that the Bank of New York's system would not be fixed that day, Mr. Bakstansky said. Lawyers began laying the groundwork for a huge overnight loan.

One issue is how such a borrowing is collateralized. Large clearing banks have agreements with customers allowing them to pledge the customers' securities as collateral under unusual circumstances. Under a separate arrangement, the Fed agrees to accept such securities as collateral. The failure to deliver trades because of computer malfunctions is "an unusual circumstance" under the agreements, Mr. Bakstansky said. It was ultimately agreed that Bank of New York's borrowings were collateralized by the underlying securities and by bank assets, he said.

The Fed kept the securities wire open all evening in the hope that the Bank of New York could gets its system operating and could complete as many trades as possible, thereby reducing its cost of funding the securities purchases overnight.

Working overtime at the New York Fed were Mr. Corrigan, the president, and officers from the legal department, open market operations, the discount window, operations, and data processing. A few blocks away, a battery of executives were at the Bank of New York's offices. Meanwhile, operations personnel were waiting at major clearing banks around the country in case the malfunction was corrected.

Shortly after midnight, Mr. Corrigan personally approved the $24 billion loan to the Bank of New York to fund the deficit in its reserve account caused by the huge surplus of securities it ended up with. The cost of the overnight loan, at a 7.5% interest rate, was estimated at $4.9 million. The securities wire was closed at 1:30 a.m., with the Fed Wire closing 45 minutes later.

At a meeting at the New York Fed Friday morning, according to one participant, Bank of New York officials told their peers the bank would be operating shortly and would not seek to "back value" transactions to Thursday — meaning the bank would bear the full cost of its systems problems.

At 11:45 a.m. Friday, the Fed suspended securities trades destined for Bank of New York "to give the bank a little breathing room," a Fed official said. Bank of New York processed Thursday's trades before starting on Friday's work. At 1:31 p.m., the Fed lifted the suspension and, once again, kept the securities wire open late — until 8:15 p.m, when the day's transactions were finally settled.

On Monday, Mr. Bakstansky said, things returned to normal. ■

American Banker December 2, 1985

Computer Failure Disrupts Treasury Market

By TOM FERRIS

The advent of high technology has revolutionized the world of money. Advanced communications networks have linked the various financial markets and allow complex transactions to be processed within seconds. This capacity, in turn, has led to an accelerated turnover of money, to the point that more than half a trillion dollars changes hands each day in the U.S. over automated systems.

Usually, these systems operate smoothly. But when a glitch develops, it can be like throwing a monkey wrench into the works.

The financial community got a taste of that 11 days ago when the Bank of New York, a key clearing agent for the purchase and sale of U.S. Treasury instruments, suffered a major computer failure.

The incident, involving the bank's link with the government securities network operated by the Federal Reserve System, prevented it from delivering more than $20 billion in instruments to their purchasers.

Because it got stuck with the securities and was unable to collect payment from the buyers, the bank suddenly found itself facing a huge cash deficit in its account at the Federal Reserve Bank of New York. Market sources said that deficit was funded by an overnight loan of $24 billion, reportedly the largest borrowing ever from the Fed's discount window.

The malfunction was corrected the following day, but it ended up being costly. Bank of New York's loss was put at $1.5 million to $2 million after taxes, which would amount to about a nickel or a dime a share. The snafu also tarnished the image of the bank, one of the largest clearers of government securities, processing more than $60 billion in trades daily.

More important, the incident displayed just how dependent on computers the financial markets have become and how vulnerable banks are to system failures, said observers, who added that there were no assurances that the same kind of failure will not beset a major clearing bank again.

Bank of New York's computer malfunction disrupted the government securities market and drove down the federal funds rate, the level of interest banks charge on loans to each other. The day after the huge overnight loan, fed funds were trading in the range of 6% to 6.5%, well below the current 8% rate. The demand for overnight borrowing was down because institutions buying securities from Bank of New York were able to hold on to the payment for another day.

Faced with an enormous cash deficit, the Bank of New York had to take a loan estimated at $24 billion from the Federal Reserve to meet its obligations.

The crisis also brought some attention to the little-publicized clearing function of the massive market for the primary and secondary trading of Treasury instruments and other government-issued securities. Virtually all of these transactions are handled through the Fed's book-entry securities system, which eliminates the need for physical certificates because it electronically records the ownership of the instruments. On an average day, securities valued at $230 billion change hands in 30,000 transactions.

The market is dominated by four large New York clearing banks, which account for more than half of the entire national volume. Along with the Bank of New York, other big players include Manufacturers Hanover Trust Co. and Irving Trust Co.

Computer difficulties at clearing banks and at the Fed itself frequently delay processing on the latter's securities wire, but the difficulties almost always are sorted out within a few hours. ■

Bank of New York Episode Catches Eyes of House Panel

American Banker

December 9, 1985

By JAY ROSENSTEIN
and BARTLETT NAYLOR

WASHINGTON — The House Banking Committee will take a harder look at the operations and supervision of the payments system in response to a Nov. 21 computer breakdown at the Bank of New York that triggered a brief crisis for the bank and for the government securities market.

The incident forced the Federal Reserve Bank of New York to lend $22.6 billion overnight to the Bank of New York, which is believed to be the largest discount window borrowing ever. The total amount of the loan was confirmed in a letter from New York Fed president E. Gerald Corrigan to the Federal Reserve Board.

In addition, according to a letter to the Fed from Congress, Manufacturers Hanover Corp. also had a computer problem the same week that resulted in more than $4 billion being borrowed by that institution.

It appears that Congress will be gathering information quite a while before legislation, if any, will surface.

Rep. Fernand J. St Germain, D-R.I., the committee chairman, last week sent a letter to Fed chairman Paul A. Volcker asking for information about the present system. Mr. St Germain noted that in the past he has offered legislation that would have enhanced federal supervision and regulation of the payments system.

And Walter E. Fauntroy, the Democratic delegate to the House from the District of Columbia who is chairman of the House Banking subcommittee on domestic monetary policy, announced that the subcommittee will hold a hearing on Thursday to probe the Bank of New York incident. Representatives from the Federal Reserve System and the Bank of New York have been asked to appear.

Mr. Fauntroy also wrote to Fed chairman Volcker asking for additional information.

The Bank of New York is the primary clearing agent for a number of major Wall Street brokers that buy and sell U.S. Treasury instruments and other government securities. Last month, following a busy day processing transactions, the bank's computerized link to the government securities network broke down and disrupted processing throughout the market.

And since the Bank of New York was unable to transfer securities and collect payment from the buyers before the end of the day, it had a deficit in its account at the New York Fed that had to be funded by the $22.6 billion overnight loan.

The reference to the $4 billion borrowing by Manufacturers Hanover appeared in a Dec. 3 letter from Mr. Fauntroy to Fed chairman Volcker. In New York, no immediate comment from Manufacturers Hanover was available.

According to the notice of the hearing from Mr. Fauntroy, the session on is intended in part to assess ways to minimize the dependence of the government securities market on just a few clearing banks; what types of additional surveillance could minimize failures; and how to handle situations in which the clearing banks' capital and assets positions are less than the outstanding deficit.

As for Mr. St Germain's concerns, he asked in a three-page letter to Mr. Volcker dated Dec. 3 if the emergency credit of $22.6 billion to the Bank of New York compares with the Continental Illinois National Bank bailout of 1984.

Mr. St Germain also asked how the problem would have undermined the banking system and national security had the Fed been unable to resolve the situation.

"I am concerned about the apparent concentration of responsibility for government securities transactions among the 35 or so 'primary dealers,' " he said. "Are we not courting Continental-scale problems by permitting financial operations of such enormous magnitude and significance to be handled by so few institutions?"

The House Banking Committee chairman also asked if the Fed has "a set of operating, security, and financial soundness standards that depository institutions must meet before they can enter into large dollar transactions."

Steve Roberts, assistant to Fed chairman Volcker, said Friday that Mr. St Germain's questions are "good" and "can't be taken lightly."

Mr. Roberts added that this week's hearings may result in some important policy statements about management of the nation's money system. ■

American Banker December 10, 1985

Corrigan, Bank of NY Chief to Testify In House Probe on Computer Failure

By TOM FERRIS

NEW YORK — E. Gerald Corrigan, president of the Federal Reserve Bank of New York, and J. Carter Bacot, chairman of the Bank of New York, can expect to face probing questions this Thursday at a House subcommittee hearing looking into a computer failure that disrupted the government securities market last month.

The House Banking subcommittee on domestic monetary policy announced Monday that the two bankers will testify on how the two-day crisis was handled and on the implications such disruptions have for the financial markets. The incident forced the Bank of New York to borrow a record $22.6 billion overnight from the New York Fed's discount window.

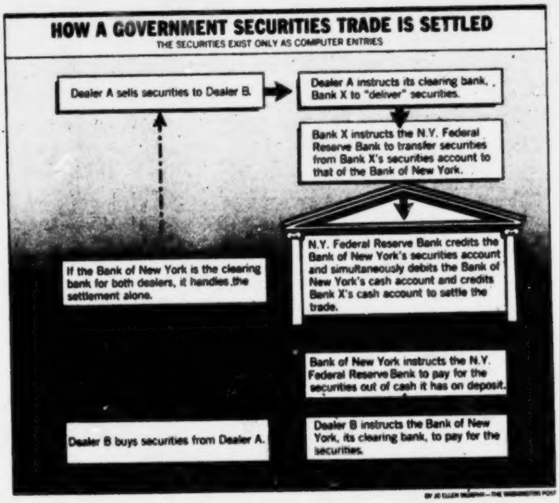
Meanwhile, the subcommittee backed away from its statement that a separate computer failure had led to a $4 billion Fed loan to the Manufacturers Hanover Corp. "The Fed has advised us that is not correct," said Howard Lee, staff director of the panel.

The chairman of the subcommittee, Walter E. Fauntroy, the Democratic delegate from the District of Columbia, wrote to the Fed last week saying he had "learned" of the loan and was concerned.

But, Mr. Lee said on Monday, the subcommittee was evidently misinformed. "We had gotten our information from sources we had believed to be reliable," he said. "If we owe Manufacturers an apology, we certainly will give them one."

On Nov. 20, the day before the Bank of New York's computer failure, the Manufacturers Hanover Trust Co. did make an overnight borrowing from the Fed, but for "nothing like the number mentioned," said James Hambleton, a senior vice president of the bank. He said that statistics released by the Fed showed that the total borrowing of all institutions that day totaled $4.1 billion.

The Bank of New York's computer failure severed the bank's link to the government securities network operated by the Federal Reserve System and prevented it from delivering more than $20 billion in instruments to their purchasers. Because the bank got stuck with the securities and was unable to collect payment from the buyers, it suddenly found itself facing a huge cash deficit in its account at the New York Fed that had to be funded by the $22.6 billion discount window loan. ■



HOW A GOVERNMENT SECURITIES TRADE IS SETTLED

THE SECURITIES EXIST ONLY AS COMPUTER ENTRIES

Dealer A sells securities to Dealer B.

Dealer A instructs its clearing bank, Bank X to "deliver" securities.

Bank X instructs the N.Y. Federal Reserve Bank to transfer securities from Bank X's securities account to that of the Bank of New York.

N.Y. Federal Reserve Bank credits the Bank of New York's securities account and simultaneously debits the Bank of New York's cash account and credits Bank X's cash account to settle the trade.

If the Bank of New York is the clearing bank for both dealers, it handles the settlement alone.

Bank of New York instructs the N.Y. Federal Reserve Bank to pay for the securities out of cash it has on deposit.

Dealer B instructs the Bank of New York, its clearing bank, to pay for the securities.

Dealer B buys securities from Dealer A.

BY JO ELLEN MURPHY—THE WASHINGTON POST

The Washington Post

Friday,

December 13, 1985

Computer Snarled N.Y. Bank

$32 Billion Overdraft Resulted From Snafu

By John M. Berry
Washington Post Staff Writer

The Bank of New York, the nation's 18th largest, had a brief $32 billion overdraft on its cash account at the New York Federal Reserve Bank when a computer failure last month snarled thousands of government securities transactions, a congressional committee was told yesterday.

By the end of the day, the overdraft had been reduced to $24 billion, and the bank actually had to borrow that amount from the New York Fed—pledging all of its assets—in order to balance its accounts overnight.

Aside from the unprecedented scale of the borrowing, and the spillover effects on the government securities market, the incident intensified concern at the Federal Reserve over the vulnerability of the nation's financial payments system to a technological glitch that could have disastrous consequences.

Federal Reserve Chairman Paul A. Volcker and New York Fed President E. Gerald Corrigan went before a House Banking subcommittee yesterday to describe how the computer failure occurred and how the Fed and the bank dealt with the crisis it caused.

On Wednesday, Nov. 20, transactions involving more than 32,000 different government securities issues poured into the Bank of New York, one of the largest processors of such deals on behalf of others.

The bank's computer system was supposed to be able to cope with up to 36,000 issues, but a programming glitch developed and, unknown to anyone, the computer began to "corrupt" the transactions and make it impossible for the bank to keep them straight.

Because of the computer system breakdown, the bank could not instruct the New York Fed where to send the securities arriving at the Fed on behalf of the bank's clients, and therefore could not get paid for them. The New York Fed was automatically taking money out of the Bank of New York's cash account to pay the sellers for the incoming securities, all of which are represented simply by computer records, rather than the familiar paper bonds still used by most corporations.

By Thursday evening, as hundreds of employes at a host of banks and government securities dealers tried to sort out the problems caused by the failure of the intricate and largely automatic network handling these transactions, the bank had a $32 billion overdraft on its cash account at the New York Federal Reserve Bank.

The bank's computer specialists finally came up with a "patch" for its computer program—a process described yesterday by its chairman, J. Carter Bacot, as the electronic equivalent of patching a tire—that allowed it to begin to clear some of the backlog. But just after midnight, the patch failed too, after the overdraft had been whittled down to about $24.2 billion.

The Fed kept both its nationwide wires for securities and cash transactions open in the early hours of Friday morning. When the patch failed, the Bank of New York was still able to borrow $700 million from other banks. The rest was covered by a $23.6 billion loan from the New York Fed. As collateral, the bank pledged all its domestic assets and all its customers' securities it was allowed to use for such purposes. Altogether, the collateral was worth $36 billion, according to the Fed.

The drama was not over. Around 5 a.m. Friday, the bank finally completed reconstruction of its customers' transactions from Wednesday. By 10 a.m., it had done the same for the Thursday deals. But, meanwhile, the rest of the government securities industry had begun its Friday activities, and securities and an overdraft were piling up again in the Bank of New York's account at the New York Fed.

"Faced with this situation," New York Fed President Corrigan told the banking subcommittee, "at about 11:30 a.m., we temporarily stopped accepting securities transfers for the account of Bank of New York in an attempt to stabilize the situation somewhat and to see whether it was practical to prevent further increases in the overdraft without causing excessive disruption in the market more generally. . . .

"Operationally, this meant that holders of government securities who had contracts to deliver those securities . . . to the Bank of New

Computer Froze Bank's Securities Transactions

York for one of its customers [in return for payment] were temporarily unable to make delivery under those contracts," Corrigan said.

The stoppage lasted only for about 90 minutes that afternoon, and news of it did not spread widely for nearly an hour. Yet that disruption at the clearing bank was enough, Corrigan said, to make some market participants unwilling to trade securities among themselves. "Perhaps most importantly, there was also some evidence that investors were beginning to seek to break trades and financing transactions with dealers serviced by the Bank of New York."

Shortly after noon, the Bank of New York was able to begin handling the Friday transactions that had been piling up, and the Fed was again able to accept securities destined for the bank. By that point, the bank was operating with a computer system that had undergone a major overhaul in less than 24 hours.

The crisis was over, but its final bill is still mounting.

The Bank of New York was out of pocket about $5 million, an amount equal to about 7 percent of its earnings in the first nine months of this year, to pay interest on the money it had to borrow that Thursday.

It is still negotiating with many of the parties who may have sustained losses in transactions that were not completed on time. Such negotiations are common, said an official of one major securities dealer, because a few transactions are always going awry. This time it was thousands.

Some customers walked away in better shape. "Indeed, those individuals and institutions who bought securities in question received a windfall in that they received interest for the day [on the securities], but did not incur any cost of financing," Corrigan noted.

But any loss or gain in dollars, even with millions of dollars worth of interest at stake, is not the real issue. What worries both Federal Reserve officials and participants in the government securities market is the potential for a failure of the system.

On an average day, about $200 billion worth of government securities transactions take place involving about 27,000 separate transactions, Corrigan said. Some days the totals are far larger.

"Like it or not," Volcker told the subcommittee, "computers and their software systems—with the possibility of mechanical or human failure—are an integral part of the payments mechanism. The scale and speed of transactions permit no other approach.

"In the last analysis, no mechanical system can be entirely 'fail-safe' and also be commercially viable," he said. "The costs would simply be too high, and the money and Treasury securities markets could not operate at the present level of efficiency."

The Fed chairman pointed out that, in this case, the Fed was available to lend the $23.6 billion, on good collateral. "The effects in this instance were of unprecedented magnitude, measured by the amount of the overnight loan," he said. "But the effects in terms of market performance and risk were well contained. . . . I believe it would be wrong to overdramatize this incident."

Corrigan in his more detailed testimony sounded more notes of concern. "I believe our actions were prudent, disciplined and appropriate. In saying this, I should also confess that in some respects were were a bit lucky," he said.

Part of the luck was that the bank was able to get its computer going again as soon as it did. Another part, Corrigan said, was that Thursday was not an especially heavy day for securities transactions.

One government securities trader summed up the situation this way:

"We're all afraid something will go bump and send the market into a tailspin. . . . The Fed is working night and day to figure out what it can do. The banks are working night and day. But the amount of [trading] in financial markets is so large that we feel this is the No. 1 financial problem of the next few months. Banks have to be able to make settlements with each other."

American Banker December 13, 1985

Volcker Downplays Loan to Bank of New York

But Fed Chairman Sees Potential for Danger to Payments System from Computer Problems

By **TOM FERRIS**

WASHINGTON — Federal Reserve Board Chairman Paul A. Volcker Thursday downplayed the significance of the central bank's record $22.6 billion overnight loan to the Bank of New York Co. last month, but he said the computer snafu that prompted the borrowing demonstrated the vulnerability of the nation's payment system to disruption and huge credit risks.

"Potentially serious implications for the payments system and the securities market were highlighted, although they were avoided in this instance," Mr. Volcker told the House Banking subcommittee on domestic monetary policy.

"I believe it would be wrong to overdramatize this incident," he said. "There was a serious operational problem which illustrated some potential vulnerabilities of the clearing mechanism. But it is also true that the problem could be dealt with effectively within our present arrangements. In that sense, the system did, in this instance, prove 'fail-safe.' "

Because the glitch involved trading of highly rated government securities, the question of collateral was not paramount and the episode remained largely an operational problem, the Fed chairman said.

However, a similar foul-up on a private system handling funds transfers, such as the New York-based Clearing House Interbank Payments System network, might lead to huge cash overdrafts — and huge risks for the banks involved and for the Fed, he added.

Mr. Volcker noted that the central bank was about to begin a program aimed at curbing the intraday overdrafts that about 1,700 banks incur on a typical day. The Fed chairman also said that banks have to balance competitive pressures to provide services at a low cost against the need for adequate backup systems and operational controls.

Testifying before the subcommittee, which has oversight authority of the Fed, Mr. Volcker and E. Gerald Corrigan, the president of the Federal Reserve Bank of New York, strongly defended the way the central bank handled the crisis and the propriety of the discount window borrowing.

"The overnight loan, huge as it was, was fully secured, with an ample margin of protection," Mr. Volcker said.

Mr. Corrigan added that although the advance was 23 times larger than the Bank of New York's capital base, the bank had pledged its customer's securities and bank assets with a total book value of $36 billion.

"Since the loan was adequately secured, the Federal Reserve Bank of New York was never subject to risk of loss," Mr. Corrigan said, "and since the capital of Bank of New York was not impaired, its solvency was not in question."

Penalty May Be Charged

However, Mr. Volcker said, in the future a special penalty assessment above the discount rate "may well be appropriate" when an exceptionally large borrowing is caused by an institution's own computer problems. Such a fee would encourage better backup systems, he said.

J. Carter Bacot, the chairman of Bank of New York, who confirmed that the overnight loans had cost his bank $5 million, bristled at that suggestion. "Five million dollars is penalty enough," he told the subcommittee. "Should there be an extra penalty? We say no."

The imposition of such a penalty might prompt the bank to reconsider whether it could afford to stay in the clearing business, Mr. Bacot said.

The Bank of New York acts as clearing agent for 40 brokers that buy and sell U.S. Treasury instruments and other government-issued securities. For a fee, it serves as the middleman, transferring ownership of the instruments to the buyer and collecting payment on behalf of the seller.

The computer snafu on Nov. 20 severed the bank's link to the Fed's book-entry government securities network. The bank's account at the New York Fed continued to be debited as securities were received from other institutions. However, the computer failure prevented the bank from sending instruments on to other banks.

As a result, the Bank of New York got stuck with more than $20 billion in instruments it could not collect payment for and found itself facing a huge cash deficit in its Fed account that had to be funded by the $22.6 billion discount window loan.

The bank was charged the 7.5% discount rate on the overnight borrowing, which subcommittee members noted was below the market rate the bank would have had to pay. In addition, the bank was charged the equivalent of about 10% interest for one day on the $1 billion deficit it ended up with in its reserve account at the close of business.

Earnings Will Be Hit

Mr. Corrigan said that the "direct out-of-pocket cost" of $5 million to the bank is "about 7% of its earnings over the first three quarters of this year, and more importantly, is a sizable percentage of the gross revenues it generates via its clearing activities."

Speaking to reporters after the hearing, Mr. Bacot characterized the impact on the bank's clearing profits as "very significant" but declined to be more specific.

One analyst has estimated that, overall, the episode will cost the bank $1.5 million to $2 million after taxes, or about a nickel or a dime per share.

The Bank of New York's problems attracted the House subcommittee's attention to the little-publicized clearing function for the trading of government securities. Virtually all of these transactions are handled through the Fed's book-entry securities system, which eliminates the need for physical certificates because it electronically records the ownership of instruments. On an average day, securities valued at $230 billion change hands in 30,000 transactions, according to the New York Fed.

The market is dominated by four large New York clearing banks, which account for more than half of the entire national volume, according to a Fed document. The Bank of New York alone is said to process an average of about $60 billion a day. Mr. Bacot said the bank holds 40% of the market share. Other big players include Manufacturers Hanover Trust Co., Irving Trust Co., Marine Midland Bank, and a New York affiliate of Security Pacific National Bank.

Walter E. Fauntroy, the delegate from the District of Columbia and chairman of the subcommittee, expressed serious reservations about the implications of the incident for the nation's financial system in a prepared statement at the beginning of the hearing.

But after hearing the testimony of the Fed officials, Mr. Fauntroy said, "I believe the Fed did take the proper actions" as far as the overnight loan was concerned.

Both Mr. Volcker and Mr. Corrigan seemed understanding of the Bank of New York's extensive computer problems, pointing out that the Fed's own system often "goes down." "Like it or not, computers and their software systems — with the possibility of mechanical or human failure — are an integral part of the payments mechanism," the Fed chairman said.

Mr. Volcker rejected a suggestion by the subcommittee that the Fed might want to enter the clearing business itself to provide stability to the marketplace. "That, frankly, would be an area we would be extremely reluctant to enter." ■

FCS Can Borrow At Discount Window, Fed Testifies

The Federal Reserve has the authority to lend through its discount window to the Farm Credit System and commercial agricultural banks, a Fed director told a congressional subcommittee.

But, FCS has not requested such credit, Stephen H. Axilrod, staff director for the Fed's monetary and financial policy, told the House Banking Committee's Economic Stabilization Subcommittee when explaining the technical aspects of possible discount-window lending to FCS.

"The Farm Credit System has considerable strengths," he said. "The amount of capital that it can bring to bear on its problems is quite large and it holds a substantial amount of liquid assets."

There appears to be enough collateral within FCS as a whole to back a substantial amount of Federal Reserve lending, Axilrod said.

The Fed discount window generally is intended to be a short-term liquidity facility for use when other sources of liquidity are not reasonably available, Axilrod said. "Borrowers are expected to show that they have made every effort to obtain needed funding from other sources, consistent with the Federal Reserve's role as the 'lender of last resort.'"

The Fed discount window is not intended to be "a facility for long-term financing or to be a substitute for risk-bearing capital," he said.

The Fed has long been authorized explicitly through the Federal Reserve Act to lend to one component of FCS — Federal Intermediate Credit Banks, Axilrod said. As a technical matter, he said, funds loaned to FICBs could be passed along to other units within FCS.

"However, the FICBs are only a small component of the system and not all of their collateral would meet the technical requirements" of the Fed, he said.

Additional authority to lend to FCS must rest on the other provisions of the Federal Reserve Act that permit loans in certain emergency circumstances to any individual, partnership or corporation, he said. Such an emergency situation requires a policy judgment and approval by five members of the Federal Reserve Board, he said.

Commercial banks active in farm lending have access to the usual discount window facilities, he said. The Fed also has a seasonal credit program for smaller depository institutions without ready access to national money markets, he said, which is intended to assist agricultural banks.

THE NEW YORK TIMES,

FRIDAY, DECEMBER 13, 1985

Fed Considers New Penalty

By ROBERT D. HERSHEY Jr.

Special to The New York Times

WASHINGTON, Dec. 12 — The Federal Reserve Board chairman, Paul A. Volcker, said today that the central bank was considering the imposition of a new, heavier penalty against financial institutions whose computer failures force them to seek exceptionally large short-term loans from the Fed.

Mr. Volcker appeared before a House subcommittee inquiring into the Nov. 21-22 episode in which the Fed had to supply $22.6 billion to the Bank of New York to prevent a possible shutdown in the huge market for United States Government securities.

The Bank of New York is the biggest clearing agent for these securities in a market that trades some $200 billion of them each day. Clearers are those who process the transfer of the securities from buyers to sellers and arrange for the related payment and collection of funds.

Mr. Volcker, citing intense competition that he said spurred severe cost cutting that could result in inadequate computer systems, held that the Fed has a responsibility to make sure there is "countervailing pressure" so that the financial structure as a whole is not subjected to unacceptably high risks from computer or other failures.

"A special penalty rate, designed to encourage better backup systems, when exceptionally large borrowing is caused by the institution's own computer problems, may well be appropriate," the Fed chairman declared. Mr. Volcker did not specify how stiff a penalty he would want.

The Bank of New York borrowed the $22.6 billion it needed at the Fed's regular discount rate, now 7½ percent. As it happened, this was more than one-half of a percentage point below the market rate that day for overnight loans in the open market.

Meanwhile, the Bank of New York chairman, J. Carter Bacot, disclosed that the computer foul-up, which was caused by a programming error, cost the bank $5 million in interest charges for the one-day loan. This unrecoverable loss, another bank official indicated, would result in a reduction of up to $3 million in fourth-quarter earnings, an amount he said would be "not material."

The bank's profits for the first nine months of 1985 were about $96 million.

'Substantial Penalty'

Taking issue with Mr. Volcker's proposal for a new levy, Mr. Bacot told the subcommittee: "I cannot see the basis for suggesting that beyond this payment of interest to the Federal Reserve some additional punitive measure should be taken against the Bank of New York. Indeed, $5 million represents a very substantial penalty for our stockholders, officers and employees."

Mr. Volcker was joined by E. Gerald Corrigan, who as president of the Federal Reserve Bank of New York led the deliberations that resulted in what was by far the biggest discount-window loan ever made by the central bank.

Mr. Corrigan said there was no alternative to granting the loan to the Bank of New York, which by 8:30 P.M. on Nov. 21 had an overdraft with the Fed of nearly $30 billion. The situation, he said, threatened "the orderly opening of business" the following day.

"The important issues raised by this episode are not so much the fact of the discount window advance but rather its terms and, even more importantly, the underlying circumstances which permitted the buildup in the overdraft in the reserve account in the first instance," Mr. Corrigan said.

While the Fed did not lose money — it actually made a modest profit — Mr. Corrigan said this was only by chance, and "leaves open the question of whether there should not be special terms associated with discount-window lending of the nature encountered in this case."